COMMODORE & AMIGA nr 7



NR INDEKSU 355216
ISSN 0867-8022

Cena 10 000 zł

lipiec 1992 r.

MAGAZYN UŻYTKOWNIKÓW KOMPUTERÓW «COMMODORE»

ZRÓB TO SAM:

* Przełącznik PAL/NTSC dla Amigi
* Zasilacz do C-64

COMMODORE 64:

* Najlepsze z najlepszych: EDYTORY TEKSTU
* Polscript 64
* Moduł MON-64
* RAPORT: Moduł MAX PL



AMIGA:

* Logic Works
* Aztec C
* Harpoon
* Od kuchni: układ Paula

... i wiele innych !



SUPERMARKET

COMMODORE 64/128

■ Sprzedam C-64, magnetofon, stację 1541, monitor mono, joysticki, FINAL II i X, programy (11 kaset, 24 gry całodyskowe). Cena: 4.9 mln. J. Ru- towski, Olszewska 22, 09-300 ŻUROMIN, tel. 72600.

■ Sprzedam C-64C, 1541II, monitor Commodore DM 602 zielony, 2 joysticki, mysz GEOS 101, 100 dyskietek (258 gier i pr. użytkowe), literaturę. Cena około 5 mln. Piotr Malec, Pionierska 13/47, 38-700 USTRZYKI DOLNE, tel. 495.

■ Sprzedam C-64, magnetofon, joystick, moduły (gwarancja) + 400 gier (2 mln) z monitorem (3.3 mln) lub za dopłatą zamienię na ATARI 520 STFM. Paweł Markowski, ul. Wilczyńskiego 22, 08-300 SOKOŁÓW PODLASKI.

■ Sprzedam do C-64: stację dysków 1541 II, monitor 1802S (kolor), 2 pudełka na 50 dyskietek 5.25", 60 dyskietek z grami. Cena do uzgodnienia. Krzysztof Czabaj, ul. Zielona 20/15, 80-746 GDAŃSK, tel. 31-12-49.

■ Sprzedam C-64 II, magnetofon, moduł PLUS, literaturę, kasety (2 mln). Piotr Mitera, ul. Orłowskiego 72/1, 32-600 OŚWIĘCIM.

■ Sprzedam C-64, magnetofon i stację 1541 II, joystick, moduły FINAL III i X, pudełko z 27 dyskietkami, 10 kaset. Cena ok. 4 mln. Sławomir Ziółkowski, Pułaskiego 7/11c m 5, 80-520 GDAŃSK.

■ Sprzedam C-64, magnetofon, moduł X, 2 joysticki, ponad 1000 programów (31 kaset), cena całości - 2.5 mln. Łukasz Biesiada, ul. Łukaszyńska 19/15, 21-500 BIAŁA PODLASKA, tel. 437-561.

■ Pilnie sprzedam C-64II z magnetofonem (gwarancja) oraz moduły, joystick, literaturę, ok. 500 gier i pr. użytkowych. Cena ok. 2.9 mln lub 340 DM. Łukasz Banaczkowski, ul. Kanałowa 5/4, 27-210 STARACHOWICE, woj. kieleckie, tel. 7233.

■ Sprzedam C-64 II, stację 1541 II, FINAL II, 45 dyskietek, literaturę za 4.0 mln. Maciej Pielech, ul. Mazurska 8/13, 08-110 SIEDLCE, tel. 290-58.

■ Sprzedam C-64, 1541 II, magnetofon 1530, 20 kaset, 23 dyskietki, ACTION REPLAY, cena 4.5 mln. Ul. Księżodworska 69, 13-200 DZIAŁDOWO, woj. ciechanowskie.

■ Sprzedam BLACK BOX III i 10 kaset z grami dla C-64 za 350000 zł. Norbert Wasiak, ul. Pobiedna 31, 26-420 NOWE MIASTO N/PILICĄ.

■ Zamienię na Amigę: C-64, stacja 1541 II (gwarancja), 50 dyskietek, magnetofon, 20 kaset, 3 moduły, joystick, literaturę w dobrym stanie lub sprzedam. A. Kondratowicz, Al. 1000-lecia 8/10, 59-700 BOLESŁAWIEC, tel. 64-74 (po 2000).

■ Sprzedam C-64II (gwarancja), magnetofon, BLACK BOX, 2 joysticki, 10 kaset, literaturę za 2.5 mln. Rafał Dobek, ul. Szarych Szeregów 3/60, 35-114 RZESZÓW.

■ Bardzo tanio sprzedam C-64 II, magnetofon, stację 1541 II, 50 dyskietek, ACTION PLUS, mysz, joystick - komplet lub osobno. Mariusz Staszak, ul. Kąkolowa 7/27, 85-811 BYDGOSZCZ, tel. 630964.

■ Sprzedam (lub zamienię na Amigę) C-64, stację dysków 1541 II, 100 dyskietek, pudełko, kilkanaście kaset, magnetofon, joysticki. Łukasz Nosal, ul. Złota 11a/8, 78-100 KOŁOBRZEG, tel. 216-28.

■ Sprzedam lub zamienię na skuter Simson C-64C magnetofon, mysz, joystick, BLACK BOX, programy i literaturę. Marek Lewandowski, ul. Wiejska 20/54, 87-800 WŁOCŁAWEK.

■ Sprzedam C-64, magnetofon, X, joystick i 800 gier w wersji kasetowej za 2.4 mln. Marcin Wojtasik, ul. Rynek 19/5, 88-150 KRUSZWICA, tel. 523.

■ Sprzedam C-64 II, stację 1541 II, magnetofon, 2 joysticki, moduły, 70 dyskietek, 30 kaset, pokrywa, pudełko - za 3 mln. Krzysztof Krzyżanowski, ul. Pionierów 4, 48-300 NYSA.

■ Sprzedam C-64 II, magnetofon, 10 kaset, literaturę. Cene 1850000 zł. Przemysław Konarski, Oś. Kopernika 4/3, 37-500 JAROSŁAW.

■ Niedrogo sprzedam C-64C, magnetofon na gwarancji, pokrywę, 3 joysticki, moduły, 23 kasety, literaturę. Jacek Nowakowski, ul. Zbójnicka 2/27, 85-791 BYDGOSZCZ.

■ Sprzedam C-64 II, stację 1541 II, magnetofon (nowe na gwarancji), moduł TURBO, joystick, 7 kaset, 10 dyskietek wraz z pudełkami, literaturę. Jacek Maciejewski, ul. Kombatantów 1, 24-140 NAŁĘCZÓW, tel. (081) 11-44-03, po godz. 1500.

■ Sprzedam C-64 II (gwarancja), Datasette 1530, 2 joysticki, moduły, pokrywę, gry, użytki, dema, opisy, mapki, literaturę, 14 kaset (365 programów). Cena 2.8 mln zł. I. Bujakiewicz, ul. Styczyńskiego 73/4, 41-500 CHORZÓW.

■ Sprzedam moduły do C-64: HELP PL (130000 + porto) i FINAL III (2500000 + porto). Rafał Czapiewski, ul. Waryńskiego 4a, 89-600 CHOJNICE.

■ Kupię stację 9900 lub zgodną z 1541 (do 1.4 mln zł). Sprzedam program do nauki jęz. ang. (dyskietka, polska instrukcja, cena 45000). Adam Kubiczek, ul. Witosa 34a/22, 40-832 KATOWICE, tel. 1501794.

■ Sprzedam stację dysków 1570 (1 mln), magnetofon (150 tys.) i monitor Neptun 156 (500 tys.). Piotr Ottenbreit, ul. Stawowa 2, 49-123 SŁAWICE K/OPOLA, tel. 340-21.

■ Sprzedam C-64 II (gwarancja), 1541 II, FINAL III, joystick, dyskietki za 4 mln. Piotr Domaradzki, ul. Parzęczewska 30/11, 95-100 ZGIERZ.

■ Sprzedam C-64-II, 1541-II (gwarancja), magnetofon, monitor mono, ponad 1200 programów za 4.5 mln. Wojciech Beśka, ul. Klonowa 13 m 33, 91-036 ŁÓDŹ, tel. 517798.

■ Sprzedam C-64 II, 1541-II (gwarancja), magnetofon, FINAL III, pokrywa, pudełko, 40 dyskietek, 20 kaset. Oskar Koseda, ul. Drogowców 21, 83-400 KOŚCIERZYNA. tel 863311.

■ Sprzedam C-128, stację 1571, magnetofon, joysticki, FINAL III i oprogramowanie na dyskietkach. Cena do ustalenia. Adrian Jużek, ul. Złota 28, 44-270 RYBNIK, tel. 26210.

■ Sprzedam C-128, magnetofon, joysticki, X, literaturę, programy, stan b.dobry. Marek Pacholski, ul. Bitwy nad Bzurą 24/36, 99-100 ŁĘCZYCA, tel. 40-59.

AMIGA 500

■ Sprzedam Amigę 500 1 MB RAM (gwarancja), modulator, 2 joysticki, 70 dyskietek. Cena: 8 mln zł. Agnieszka Dybowska-Dyk, Wrocławska 20c/25, 45-707 OPOLE.

■ Sprzedam niedrogo roczną Amigę 500 (1 MB), modulator TV, monitor 1084S (gwarancja), literaturę, czasopisma, dyskietki. Razem lub osobno. Krzysztof Suchomski, Kijaszkowo 23, 89-321 TŁUKOMY, woj. pilskie.

■ Sprzedam dyskietki do A500 z programami Matematyka, Funkcje, Baza danych, Ortografia, Piórko (z opisami). Cena kompletu ok. 350000 zł. Daniel Szmańda, 88-320 STRZELNO, Ciechrz 10, woj. bydgos- kie.

■ Zamienię PC XT 640 KB, 2x360 FDD, Hercules, osprzęt, programy + 1 mln na Amigę 500. Irek Zawadka, Oś. Miranda 1, 62-700 TUREK, tel. 48-70.

■ Sprzedam A500 (1 MB RAM, modulator), monitor kolor PHILIPS 8833, 100 dyskietek, disk box. Gwarancja. Cena 11 mln. Robert Miluk, ul. Bosmańska 31a/27, 75-257 KOSZALIN.

OPROGRAMOWANIE

■ Oferuję ponad 500 gier, dem i programów dla C-64/128 na dyskietkach i kasetach. Rozprowadzam gry firmy GRUBCIO. Marcin Chrapusta, Pułaskiego 30/30, 33-100 TARNÓW.

■ Poszukuję programów (mile widziane opisy) współpracujących z drukarką LC-20 i C-64 (magnetofon). Piotr Szybiak, Osiedle Orła Białego 1a/19, 62-200 GNIEZNO.

■ Wymienię oprogramowanie, opisy, mapy. Posiadam C-64 + magnetofon. Konrad Szajner, Wąska 16/46, 22-400 ZAMOŚĆ.

(DOKOŃCZENIE ORAZ AKTUALNY KUPON PUBLIKUJEMY NA STRONIE 7).

Parę dni temu miałem okazję przysłuchiwać się biadoleniu w środkach masowego przekazu nad stanem ogólnym motoryzacji rodzimej. Z całości można było wysnuć wniosek, że do jej nędznego stanu przyczynił się głównie prywatny import używanych samochodów zachodnich oraz kompletny brak patriotyzmu i zrozumienia wśród rodaków; wolą oni (niestety!) jeździć zagranicznym wozem wyeksploatowanym zamiast nowym krajowym.

Kolejnym hitem jest cło na sprzęt komputerowy sprowadzany z Azji. Po interwencji EWG okazało się, że sprzęt ten produkowany głównie w garażach to tandeta i jeśli chcesz do kraju przywieźć tandetę to musisz bulić. Tu jednak sprawa jest nieco bardziej skomplikowana, gdyż marek, firm i firmiątek produkujących „składaki" jest sporo.

W celu wyeliminowania oszustwa każdy producent powinien dołączać do sprzętu odpowiedni „paszport" informujący, że dany komputer został wyprodukowany właśnie w Europie i w związku z tym jest z cła zwolniony. O ile wiem wszystkie komputery Commodore są zwolnione z cła i wyposażone w taki certyfikat — od pecetów zacząwszy a na Amidze i C-64 skończywszy.

Na drodze do całkowitego szczęścia (czyjego?) stoi jednak obawa, że stan ten z dnia na dzień może ulec zmianie, gdyż, jak słuchy chodzą, część niedoszłych bankrutów wzięła się w zamian do fabrykowania wymaganych paszportów. Z dobrze poinformowanych źródeł dochodzą wieści, że obecnie biznes ten jest znacznie bardziej opłacalny aniżeli produkcja samych pecetów.

KLAUDIUSZ DYBOWSKI

M • E • N • U

magazyn użytkowników komputerów «COMMODORE»



Redaktor naczelny: KLAUDIUSZ DYBOWSKI
Sekretarz redakcji: CHRISTIAN GRZENKOWICZ
Opracowanie graficzne: JOLANTA PRZEŹDZIECKA
Redaktor techniczny: JOLANTA SZUMOWSKA
Zdjęcia: JERZY STOKOWSKI
Stali współpracownicy: ANDRZEJ BOBEK (szef Działu Amigi), BARTŁOMIEJ DRAMCZYK, JERZY DUDEK, MARIUSZ FERDYN, BARTŁOMIEJ KACHNIARZ, WOJCIECH KAZIMIERCZAK, PIOTR LISZEWSKI

Redakcja: RAFAŁ PIASEK, BARTOSZ SMAGA, RAFAŁ WIOSNA, ul. Wasilkowskiego 7, 02-776 WARSZAWA, tel./BBS: 643-1840
Kontakt z Czytelnikami: pon-pt w godzinach 10.00–17.00
Wydawca: Spółdzielnia „Bajtek" ul. Wspólna 61 00-687 Warszawa tel./fax: 21-12-05
Skład i druk: Przedsiębiorstwo Wydawniczo-Poligraficzne „GRYF", S.A. Ciechanów
Korekta: KRYSTYNA WYDURSKA, MARIA GOŹDZIEWSKA
Nr zlecenia: 30852
Nakład: 72 tys. egzemplarzy

AMIGA TOTAL

Książka ta została wydana w bardzo podobnym stylu, jak C-64 TOTAL. Z tej samej serii na księgarskich półkach świeci tytuł IBM TOTAL. Nie zachęcam jednak do kupna, bo tak naprawdę nic tam nie znajdziecie.

Pierwsza część AMIGA TOTAL może odstraszyć każdego Czytelnika. Dlaczego? No bo po co komu dwa niemal identyczne podręczniki do obsługi AMIGA DOS i całego systemu wraz ze szczegółową instrukcją w stylu: „... naciskając lewy przycisk myszy na górnej linii danego okna (i trzymając go), okno można przesuwać w dowolnym kierunku. Okna nie można wysunąć poza ekran, na którym aktualnie pracujemy. Puszczając lewy przycisk myszy ...” Nie twierdzę, że nie znajdziesz tu czegoś dla siebie, lecz większość informacji dubluje się jednak z podręcznikiem do AMIGI.

Oko początkującego programisty spocznie dopiero na drugiej części AMIGA TOTAL. Nie jest to bynajmniej rozdział dla zaawansowanych. Tutaj można znaleźć go. proste programy tłumaczące korzystanie z biblioteki intuition z poziomu języka C, asemblera a nawet Amiga-Basic. Wszystko uzupełnione jest wieloma przykładami, co pozwala na pokonanie nawet bariery językowej. Proste procedury graficzne, obsługa okien, programy przeprowadzające konwersję z asemblera na linie DATA w BASIC — to wszystko można znaleźć w drugiej części AMIGA TOTAL. Myślę, że warto przekazać tę książkę początkującym.



AMIGA TOTAL
Markt & Technik 1991
CENA: 51 000 ZŁ

Trzecia część, o szumnej nazwie AMIGA UND VIDEO to gratka dla ludzi zajmujących się grafiką. Niestety nie ma tutaj np. opisu programu IMAGINE 2.0, czy REAL 3D, jednak można zapoznać się z wieloma innymi, mało znanymi na polskim rynku. Można tutaj zapoznać się z obsługą świetnego programu „TRICK STUDIO” napisanego przez programistów wydawnictwa Markt & Technik. Taka jest właśnie trzecia część AMIGA TOTAL — jest to wprowadzenie w świat animacji. Pomimo faktu, że rozdział ten przejrzałem dosyć pobieżnie muszę pogratulować autorom sposobu przedstawienia tematu.

Wydaje mi się, że AMIGA TOTAL nie jest tak wartościowym dziełem jak niedawno opisywana książka C-64 TOTAL. Bez względu na wszystko: 49 DM, a w Polsce 51 tys. zł za dość potężną dawkę wiedzy wydaje mi się korzystną transakcją dla Czytelnika.

BARTŁOMIEJ DRAMCZYK

Zamierzam nabyć komputer marki Commodore Amiga 500. (...)
1. Jaka jest różnica między klawiaturą angielską, a niemiecką?
2. Czy w pudełku, w którym znajduje się Amiga będzie myszka?
3. Czy myszka jest niezbędna do pracy z Amigą i czy można zastąpić ją joystickiem?
4. Czy gry na Amigę mogą być zarażone wirusem i jakie szkody wirus może wyrządzić komputerowi?
Krzysztof Leśniewski, Gliwice

Różnica między klawiaturami angielską i niemiecką jest niewielka — wersja niemiecka ma dwa dodatkowe klawisze dla znaków, które normalnie nie zmieściły się na klawiszach po dodaniu znaków niemieckich. Wszystkie niemal programy przyjmują, że komputer ma klawiaturę angielską i tak też interpretują wciśnięte klawisze. Powoduje to konieczność zapamiętania, który co oznacza, gdyż napisy na niej nie odpowiadają znakom z klawiatury anglojęzycznej. Definicję klawiatury można zmienić programowo (tylko wtedy, gdy dostępny jest system operacyjny). Jedynym znanym mi programem wykorzystującym dwa dodatkowe klawisze jest Protracker.
2. Oczywiście, do każdej Amigi dołączana jest (a przynajmniej powinna być) myszka. Ze słyszenia znam jednak kilka opowieści o cwaniaczkach, którzy usiłowali sprzedawać myszki „osobno”.
3. Z myszką jak z klawiaturą — teoretycznie można się bez niej obejść, ale w systemie zorientowanym graficznie (okna, ikony, itp.) utrudnia to niesłychanie pracę. Natomiast zamiana myszy na joystick jest możliwa dzięki specjalnemu oprogramowaniu, ale powstaje pytanie — czy jest sens? Trudniej będzie operować kursorem, ponadto w standardowym joysticku jest zwykle tylko jeden rozpoznawany przycisk...
4. Tak jak wszystkie dyski, gry też mogą być zarażone wirusem. Jednak gdy gra nie korzysta z systemu operacyjnego (np. gdy ma swój własny bootblock), to po infekcji zwykle nie będzie jej można uruchomić (lub też wirus będzie powodował blokowanie się komputera). W większości wypadków jednak gra sama niszczy wirusa kasując go w pamięci.
Gry korzystające z systemu będą działały, ale dyskietka taka będzie potencjalną bombą bakteriologiczną. Na szczęście gier wymagających na rozmnażanie się wirusa) jest mało.

jrme

Z tym dyskiem twardym dla Commodore 64, który podaliście w swojej ankiecie w numerze 04/92 to bujda, prawda?

Oczywiście, że NIE! Kilka lat temu firma XETEX wyprodukowała dwa typy takich urządzeń dla C-64 i C-128. Oba dyski miały pojemność 20 MB i były zorganizowane w sposób podobny do „pecetów” tzn. miały np. możliwość tworzenia podkatalogów. Urządzenia te zostały opisane w numerze specjalnym o Commodore z roku 1987, w artykule zatytułowanym „Leutenant Kernal” (ponieważ to była ich oficjalna nazwa).
Główną wadą obu dysków była ich cena, bardzo bliska tysiąca dolarów, co stawiało pod bardzo dużym znakiem zapytania opłacalność całego interesu. W tamtym okresie czasu za tę sumę można było kupić Amigę 1000 czy PC/AT...

(KAD)



ACTION REPLAY MK III

– drugie spojrzenie

Tak się złożyło, że miałem możność przez kilka dni zapoznać się z AR MK II, zanim stałem się właścicielem AR MK III. Czy i jakie różnice występują pomiędzy nimi? Oto garść spostrzeżeń, dotyczących wpisywania poleceń bez wdawania się w szczegóły, bo od tego jest instrukcja obsługi.

Zewnętrznie, poza kolorystyką napisów nie widać różnic. Ukryte są one wewnątrz obudowy. Systemowy ROM w MK III został powiększony dwukrotnie i liczy sobie 256 KB. Za te same pieniądze użytkownik otrzymuje dodatkowo:

- wbudowany na stałe w system kopier BURST NIBBLER,
- handler joysticka,
- opcję memory watch points,
- deep trainer,
- obsługę drukarki,
- zwiększoną liczbę bezpośrednio dostępnych z systemu poleceń CLI
- rozbudowane graficznie (dwie strony) okno Preferences.

Wszystkie opcje, występujące w AR MK II są dostępne w AR MK III. Z tego powodu zajmę się jedynie powyżej wymienionymi nowościami.

Programu BURST NIBBLER omawiać nie trzeba, gdyż jest on dobrze znany użytkownikom A500. W wersji przystosowanej do MK III obsługa jest identyczna jak programu wgrywanego z dyskietki. Wywołanie programu z poziomu systemu działa bez zastrzeżeń. Niestety druga możliwość, o której wspomina instrukcja — nie działa i sytuacji tej nie zmienia ustawienie w oknie Preferences opcji fast nibbler. Ponadto wyjście z kopiera przez kliknięcie na Quit powoduje zawieszenie się systemu i dopiero reset z klawiatury ratuje nas z opresji. Jeżeli już (co ostatecznie można zrozumieć) system musi się „wieszać", to nie widzę przeszkód aby Quit nie mogło załatwić od razu inicjalizacji, jeżeli w ogóle ma służyć do czegokolwiek poza ozdobą.

W programach, które obsługuje się za pomocą dużej liczby klawiszy, mamy możliwość wykonywania tej czynności joystickiem. Służy temu opcja megastick. I znów, niestety, jej uruchomienie z poziomu systemu jest prawidłowe i mamy możliwość zdefiniowania, jakim kombinacjom położeń joysticka (łącznie z „fire") odpowiadają poszczególne klawisze, ale jej wywołanie przez aktywację odpowiedniego okna w Preferences kończy się fiaskiem. Na szczęście opcja SAVE zdefiniowanych położeń joysticka, wywoływana z Preferences działa bez zarzutu. Wyjście z opcji z poziomu systemu nie przedstawiało problemu i działa skutecznie.

Opcja safedisk (uruchamia się ją z poziomu systemu) ratuje nas w sytuacjach, gdy mamy na dysku błędy w programie (poza hard error), które normalnie powodują pojawienie się komunikatów System Error lub Guru.

Opcja ta ma możliwość wybrania kilku podopcji lub uruchomienia ich wszystkich jednocześnie. Na uwagę zasługuje możliwość usunięcia denerwującego stukania głowic w stacji dysków. Również i w tym przypadku okienko safedisk w Preferences służy tylko do ozdoby. Dzięki safedisk uruchomiłem kilka, wydawałoby się kompletnie „padniętych" programów. Niesamowita opcja!

Memory watch points jest przeciekawą opcją dla „grzebaczy" w programach. Umożliwia ona odczytanie zawartości wszystkich (łącznie z PC) rejestrów w momencie, gdy zawartość wybranej komórki pamięci ulega zmianie. Stanowi ona wspaniałe uzupełnienie pułapek (break points) przy testowaniu i badaniu programu.

Opcja trace pozwala na odczyt zawartości wszystkich rejestrów w chwili przerwania programu przyciskiem freeze. Istnieje możliwość wyboru sposobu traktowania przez trace podprogramów. Mogą one być uznane za jedną instrukcję bądź też mogą być wykonywane. O przydatności opcji dla „grzebaczy" nie trzeba chyba przekonywać.

Deep trainer to wymarzone narzędzie dla bawiących się w unieśmiertelnianie gier. Jego uruchomienie wymaga jednak posiadania 1 MB pamięci, a także, poza wykonaniem czynności opisanych w instrukcji — wykonania resetu z klawiatury. Instrukcja o tym nie wspomina, a jest to niezbędny warunek uaktywnienia deep trainera! Ostrzegam jednak tych, którzy sądzą, że zastąpi ona rozum lub znajomość asemblera procesora 68000. Bardzo rzadko uda się sztuka unieśmiertelniania gry metodą „wprost", na podstawie jedynie dostępnych w systemie instrukcji. Z reguły nie obchodzi się bez analizy programu i skorzystania z szerokich możliwości oferowanych przez instrukcje monitora pamięci. Jest również kwestią otwartą czy deep trainer będzie działał po podłączeniu do bottom slot zalecanego przez Commodore rozszerzenia o 1,5 MB FAST RAM, co przy pewnej przeróbce hardwaru (jeden przełącznik) udostępnia posiadaczowi 1 MB CHIP i 1 MB FAST RAM. Otwierałoby to bardzo interesującą możliwość unieśmiertelniania gier wymagających 1 MB pamięci, a jest ich coraz więcej. Niebawem będę posiadał takie rozszerzenie i poinformuję Czytelników, czy i jak współpracuje ono z deep trainerem.

AR MK III posiada nieco więcej dostępnych z poziomu systemu poleceń CLI, jak: rename, relabel, copy, dcopy. Ogólnie działają one bez zastrzeżeń, ale jeżeli będziemy np. działać na plikach bez podania ich nazwy, to pojawi się requester przypo-



minający nam o popełnionym nietakcie. Uważam, że najcenniejsze jest copy, umożliwiające kopiowanie plików bez wgrywania Disk Mastera lub CLI-Mate.

Opcja obsługi drukarki wygląda na bardzo ubogą. Piszę „wygląda", gdyż nie miałem szansy sprawdzenia działania ustawiania wielkości czcionki. Wydruk tekstu, sądząc z instrukcji, prawdopodobnie odbywa się na zasadzie WYSIWYG, ale podstawowe jest pytanie, jakie drukarki akceptuje handler MK III. Na razie pozostaje ono bez odpowiedzi. Co by tam nie było „upchane", nam i tak pozostaje korzystanie ze spolszczonego, choć pirackiego CEDa lub Scribble'a.

Nie ulega wątpliwości, że MK III jest produktem lepszym, o większych możliwościach niż MK II, ale na pytanie, czy MK III to jest to, odpowiedź brzmi: prawie. Prawie, bo freeze nie działa w 100%. Okazuje się, że gry firmy Delphine Software są odporne. Program wprawdzie restartuje po freeze, ale przestaje reagować na naciskanie klawiszy myszki, chociaż jej kursor porusza się. Sytuacji nie poprawiają, ponoć temu służące, opcje test1 i test2, dostępne z Preferences. Widać, że uruchamiają się, ale niestety bez rezultatu.

Pozostałe, identyczne jak w AR MK II opcje działały bez zarzutu, ale trzeba zauważyć, że powyższy opis siłą rzeczy dotyczy zachowań tylko mojego egzemplarza i nie można wykluczyć, że inne będą zachowywać się odmiennie. A nie jest to czcza teoria, gdyż posiadany AR MK II zwróciłem, ponieważ odmawiał on powrotu do systemu po zakończeniu pracy w opcjach przeszukujących pamięć. Nawiasem mówiąc, widziałem AR MK II bez tej wady. W każdym razie MK III jest jej pozbawiony.

Opracowanie nowej wersji trwało zaledwie pół roku. Teraz pozostaje czekać na wersję MK IV, w której wszystkie błędy bądź kłopotliwa obsługa i uruchamianie zostaną usunięte i poprawione.

Moim zdaniem w MK IV przydałby się solidniejszy monitor dyskowy, wzbogacony o opcję search. Myślę, że byłoby też możliwe wprowadzenie pracy krokowej procesora, która poza zastosowaniem do testowania programów mogłaby być użyta w super trainerze. Wtedy autorom pozostałoby jedynie wbudowanie fontanny. Żart? Nie — programowo jest to możliwe! Freeze, klawisz F i już widzę, jak rzedną miny posiadaczy innego 16. bitowca. Powtórzmy (zmodyfikowaną) mądrość ludową: dobry żart 59^{99} funtów wart! o czym, mam nadzieję, Was przekonałem.

URAN

LOGIC WORKS V2.01

Logic Works jest jedynym programem służącym do symulacji pracy obwodów cyfrowych, jaki mi się udało znaleźć dla Amigi. Logic Works, pomimo wielu wad, może być nieocenioną pomocą dla konstruktorów-elektroników. Zamiast montować układ próbny na płytce i testować jego działanie, wystarczy narysować jego schemat i... eksperymentować. Niczego tu nie można zepsuć, co najwyżej osiągnięty efekt będzie inny od oczekiwanego.

O zaletach takiego rozwiązania nie trzeba chyba nikogo przekonywać. W programie zawarte są biblioteki z gotowymi elementami. Można tu wymienić np. Buff-Reg Library, w której mamy takie elementy, jak rejestry przesuwne, bramki ośmiobitowe i wiele różnych buforów. Z kolei Counters Library zawiera cały asortyment liczników, począwszy od asynchronicznych dziesiętnych (7490) poprzez modulo 12 (7492), a na synchronicznych binarnych (74193) skończywszy. Dec-Mux-Arith Library zawiera multipleksery, demultipleksery, układy arytmetyczne itp.

Jedną z ciekawszych bibliotek jest LSI Library. Umieszczono tu np. procesor Z80 oraz pamięci EPROM. Logic Works byłby naprawdę dobrym programem, gdyby... symulował pracę tych układów. Niestety, ta biblioteka, jak i następne, o których jest mowa poniżej, mogą służyć tylko do ładnego i szybkiego rysowania schematów. Jest to podstawowa wada tego programu. LOGIC WORKS nie symuluje również pracy układów analogowych, pomimo zawartej w nim biblioteki Analog IC Library. Umieszczono w niej np. układ timera 555, transoptor i komparator napięcia itp.

Również Transistor Library i Diode Library będą nam pomocne tylko przy rysowaniu schematów. Dioda LED, podłączona do wyjścia inwertera nie świeci, pomimo obecności napięcia na tym wyjściu. W RCL Library znalazłem rezonator kwarcowy, więc z zapałem zabrałem się do rysowania schematu oscylatora z tymże rezonatorem. I tu, jak się spodziewałem, spotkało mnie rozczarowanie. Próbnik stanów logicznych, zamiast ciągu zmieniających się na przemian zer i jedynek, pokazuje stan niezidentyfikowany (X). Dlatego nie należy się przejmować, jeżeli narysowany przez nas układ nie działa na symulatorze — nie oznacza to wcale, że nie będzie on działał w praktyce. Jeśli schemat zawiera elementy pracujące poza zakresem TTL, to program będzie wykazywał stan nieoznaczony (X) lub stan wysokiej impedancji (Z). W rozwijanym menu mamy szereg podstawowych elementów techniki cyfrowej, takich jak bramki AND, NAND, OR, NOR, NOT, XOR i podobne (oczywiście dwu-, trzy-, cztero- i ośmiowejściowe), bufory, sumatory, rejestry przesuwne i wiele innych. Z ciekawostek należy wymienić: wyświetlacz cyfr dziesiętnych przychodzących do jego wejścia w postaci czterech bitów, klawiaturkę numeryczną, na której czterobitowym wyjściu znajduje się wybrana cyfra, próbnik stanów logicznych i przełącznik, służący do szybkiej zmiany stanu logicznego. W menu znajduje się także generator zegarowy (Clock Generator), który należy umieszczać wszędzie tam, gdzie potrzebujemy fali prostokątnej. Narysowanie oscylatora w postaci np. dwóch inwerterów spowoduje przypadek opisany powyżej (niedziałanie układu).

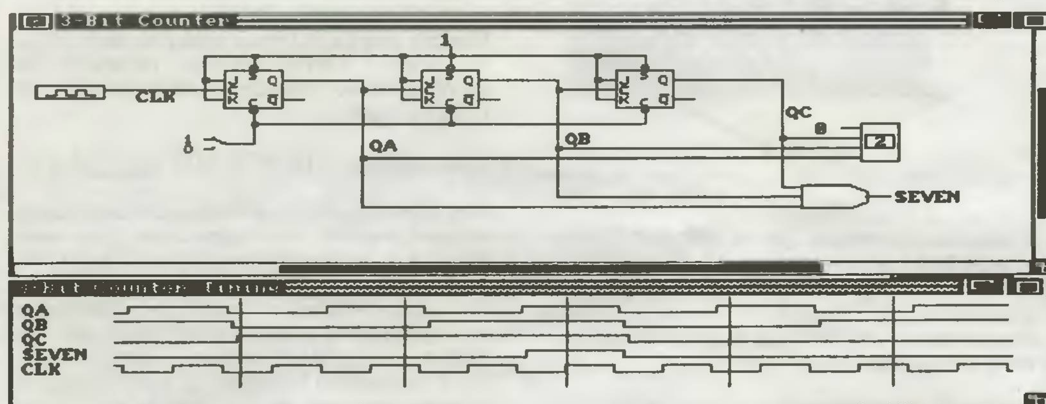
Sporą zaletą programu jest wykreślanie przebiegów czasowych (Timing Diagrams) układu. Ułatwia to bardzo śledzenie pracy obwodu. Zarówno przebieg, jak i schemat możemy wydrukować na

drukarce. Możliwa jest także modyfikacja parametrów wyświetlania diagramu (np. ilość taktów zegara na jednym ekranie, odległość pomiędzy interwałami itp.) LOGIC WORKS, oprócz symulacji pracy złożonych układów cyfrowych, pozwala też na zapoznanie się z techniką cyfrową początkującym. Możemy np. prześledzić działanie prostej dwuwejściowej bramki AND (rysunek 3), na dyskietce znajduje się także program LOGIC WORKS DEVICE EDITOR (rysunek 2). Umożliwia on dołączanie do istniejących już bibliotek nowych, definiowanych przez użytkownika elementów. Pomimo opisywanych już wad program może być bardzo pomocny przy projektowaniu układów cyfrowych.

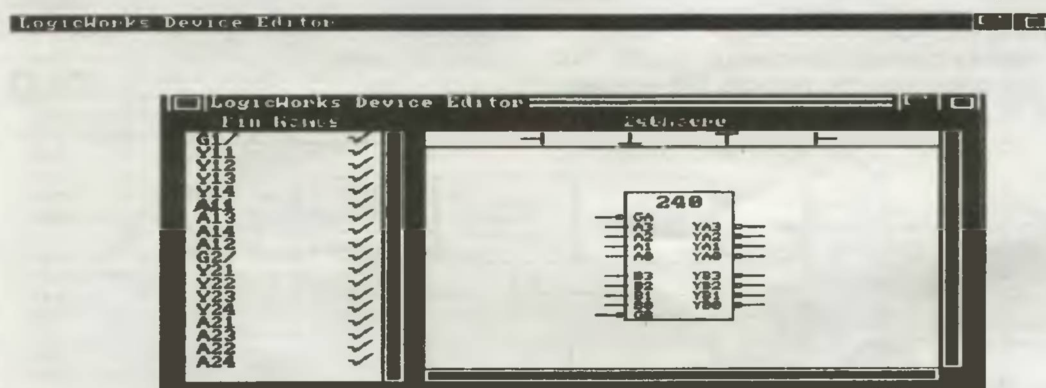
W chwili obecnej staramy się o uzyskanie najnowszej wersji programu; istnieje więc szansa, że jego dokładny opis zajmie którąś ze stron przeznaczoną na redakcyjne raporty.

JERZY DUDEK

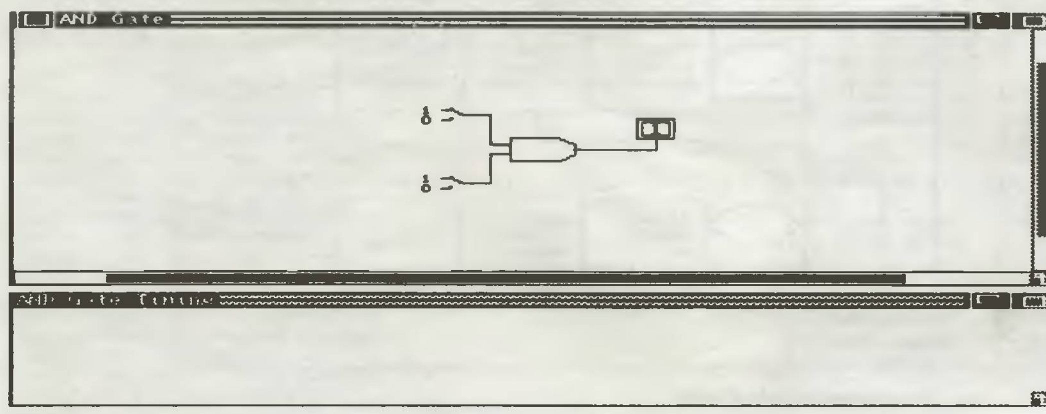
Rys. 1



Rys. 2



Rys. 3



■ Nawiążę kontakt z uczącymi się asemblera 6510 na C-64 w celu wymiany doświadczeń. Jan Kubik, Podhalańska 11/29, 41-907 BYTOM.

■ Sprzedam ok. 300 gier na C-64 (kasety) za 250000 zł. Sebastian Wojcieszek, Zielna 9/100, 26-110 SKARŻYSKO-KAMIENNA, telefon 51-30-96.

■ Kupię/wymienię programy dla C-64 oraz Bajtki 8/91 i 9/91. Marcin Książak, ul.Prosta 8b/4, 07-200 WYSZKÓW.

■ Wymienię lub sprzedam gry komputerowe. Mam C-64 i magnetofon. Marcin Dudek, ul. Dzierzby 12/1, 02-836 WARSZAWA-PYRY.

■ C-64 - oferujemy ponad 600 gier. Katalog (10000 + koperta i znaczek): Robert Ścibak, Nowa Wieś 30, 22-417 STARY ZAMOŚĆ, woj. zamojskie.

■ Wymienię / odkupię oprogramowanie do C-64. W. Kukliński, Łęgajny 8B, 11-010 BARCZEWO.

■ Wymienię programy dla C-64 (kasety). Wojciech Szywalski, ul. Konopnickiej 15, 32-200 MIECHÓW.

■ Wymienię programy użytkowe na C-64 (stacja). Tomasz Szymik, ul. Działkowa 23, 42-750 KALETY.

■ Amiga 500 - odkupię pełny opis i instrukcje w języku polskim do gier: F-19 Stealth Fighter, Flight of the Intruder, Birds of Prey. Tomasz Dubiecki, ul. Świdnicka 92/2, 57-401 NOWA RUDA.

■ Wymienię gry i użytki dla C-64, magn., stacja). Szukam programów do robienia dem (Intro Designer). Michał Cajsel, Gorkiego 53/26, 92-519 ŁÓDŹ.

■ Wymienię programy (taśma, dysk), doświadczenia, literaturę, czasopisma, schematy, innowacje, pomysły, rozszerzenia RAM dla VC-20, C-64/128/-/16/116/+4. Informacja: koperta i znaczek. COMMODORMAN, 46-058 FOSOWSKIE, skr. poczt. 3.

■ Poszukuję programów dla C-64:Last Ninja II, Last Ninja III, Saboteur I, Game Over I. Nie wiem jak uśpić smoka kulkami w Last Ninja I. Wymienię opisy do gier. Marek Dziewulski, ul. BWP 10, 21-400 ŁUKÓW, woj. siedleckie.

■ Sprzedam: Bajtki, układ scalony ADC 0809, pomogę w zrobieniu samplera do Amigi / zrobię na zamówienie. Informacja: koperta+znaczek, Radosław Sapieja, Gliniana 77/1, 50-526 WROCŁAW.

■ Kupię książkę J. Kostrzewskiego "Motorola 68000 - lista rozkazów mikroprocesora" oraz inne pozycje z dziedziny asemblera, języka C i kodu maszynowego na Amigę 500. Krzysztof Górnik, 43-414 PRUCHNA 231.



AMIGA od kuchni: PAULA

Kolejnym układem specjalizowanym zamontowanym w Amidze jest Paula. Jej głównym zadaniem jest kontrola operacji we/wy. Paula obsługuje takie urządzenia jak stacja dysków, łącze szeregowe RS-232, ponadto jest wykorzystana do obsługi generacji dźwięku oraz odczytu tzw. wiosełek (paddle). Oprócz tego Paula obsługuje cały system przerwań.

Z czternastu możliwych źródeł Paula generuje sygnały przerwań dla procesora. Przerwania na poziomach 1–6 są wytwarzane na linii IPL procesora. Paula daje programiście możliwość zezwolenia lub zabronienia przerwań wszystkich źródeł.

Transmisja danych do i ze stacji dysków jak również generacja dźwięku przebiegają z udziałem DMA. W czasie odczytu z dyskietki nie zawsze jest możliwe sprawdzenie czy następne słowo jest gotowe do transmisji DMA za pośrednictwem układu Agnus. Powodem tego jest zmienna prędkość odczytu. Również przy tworzeniu dźwięku Agnus nie może przewidzieć odpowiedniego momentu dla transmisji kolejnych danych. Aby uzyskać płynną transmisję DMA wyposażono Paulę w linię DMAL informującą Agnus o konieczności DMA.

Układy dla transmisji szeregowej są zawarte wewnątrz bloku o nazwie UART. Jest to skrót od angielskiej nazwy Universal Asynchronous Receive Transmit, co można przetłumaczyć jako uniwersalny, asynchroniczny nadajnik/odbiornik.

Znaczenie poszczególnych linii:

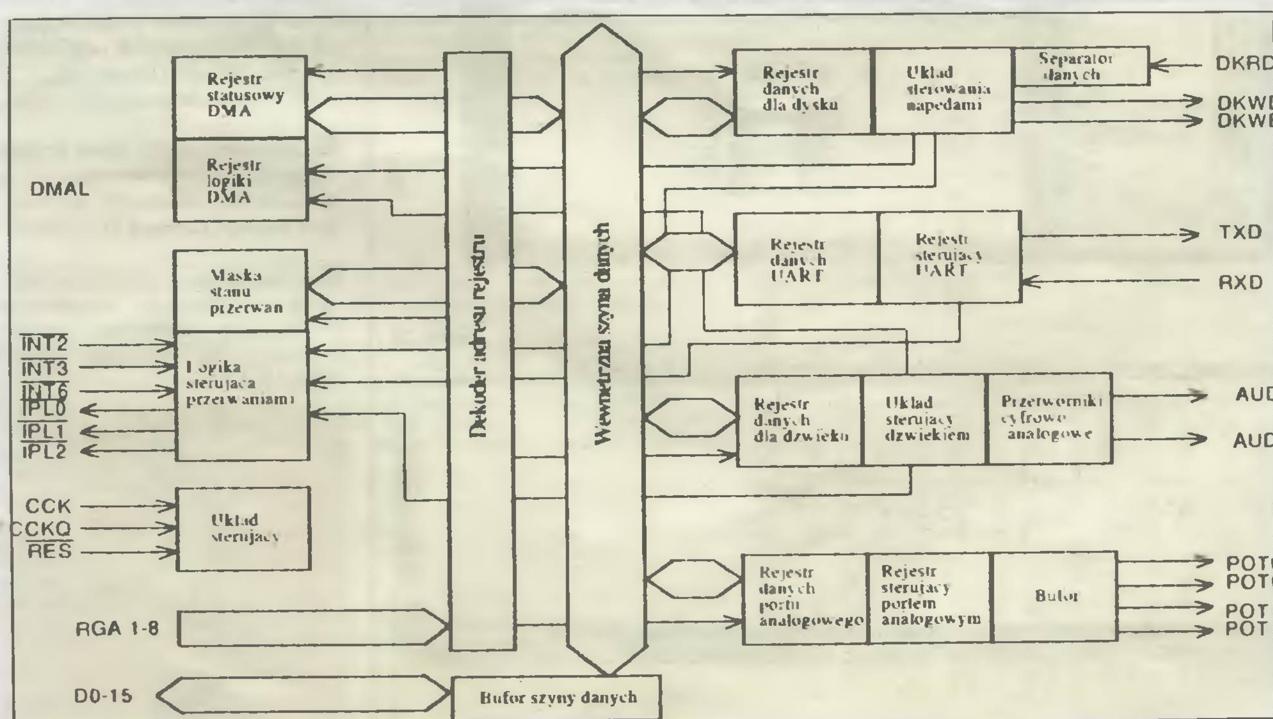
- Szyna danych D0–D15
  Tak jak w innych układach Amigi jest podłączona do szyny danych CHIP RAM.
- Szyna adresowa rejestrów RGA 1–8
  Ta szyna jest tylko wejściem. Wybiera wewnętrzny rejestr przy pomocy dekodera adresu rejestru.
- Sygnały zegarowe CCK i CCKQ



Znaczenie identyczne jak w układzie Agnus (patrz „C&A" 06/92).

- Sygnał RES
  Sygnał ustawiający Paulę we wcześniej zdefiniowany stan początkowy.
- Sygnały AUDL i AUDR
  Wyjścia dźwięku dla kanału lewego i prawego. Do linii AUDL są podłączone kanały 0 i 3, do AUDR — 1 i 2.
- Linie TXD i RXD
  Sygnały interfejsu szeregowego RS-232. RXD jest to linia odbiornika (wejście), TXD — nadajnika (wyjście). Linie te pracują w standardzie TTL, jednak między Paulą i gniazdem łącza szeregowego umieszczono konwerter napięć do poziomu spotykanego w RS-232.
- Wejścia analogowe POT0X, POT0Y, POT1X i POT1Y
  Sygnały POT0X i POT0Y są podłączone do gniazda joysticka/myszy o numerze 0, natomiast POT1X i POT1Y — o numerze 1. Do tych gniazd możemy podłączać wcześniej wymienione wiosełka oraz joysticki analogowe, zbudowane w oparciu o potencjometry. Paula odczytuje wartość rezystancji danego potencjometru i wartość tę zapisuje w rejestrze odpowiadającym danemu potencjometrowi. Wszystkie cztery wejścia analogowe mogą być programowo ustawione jako wyjścia.
- Linie DKRD, DKWD i DKWE
  Poprzez linię DKRD Paula odczytuje dane z dysku, poprzez DKWD zapisuje, natomiast linia DKWE służy do przełączenia stacji dysków z odczytu na zapis.
- Linie przerwań INT2, INT3, INT6 oraz IPL0, IPL1, IPL2
  Trzy linie INT informują Paulę o poziomie przerwania. Linia INT2 jest normalnie podłączona do układu CIA-A. Jest także podłączona do interfejsu RS-232 oraz szyny rozszerzeń (Expansion Port). Linia INT3 jest podłączona do wyjścia układu Agnus o tej samej nazwie, natomiast linia INT6 do układu CIA-B oraz szyny rozszerzeń. Linie o pozostałych numerach są wykorzystane do obsługi urządzeń we/wy sterowanych przez Paulę i nie są wyprowadzone na zewnątrz układu. Linie IPL0—IPL2 są podłączone bezpośrednio do odpowiednich linii procesora i służą do generacji przerwania procesora na odpowiednim poziomie.

DUD



Struktura układu Paula



Rejestry układu PAULA

Poniżej przedstawiam wykaz rejestrów znajdujących się w układzie PAULA. Jeżeli chcemy obliczyć adres efektywny interesującego nas rejestru, musimy do adresu z tabeli dodać adres bazowy obszaru rejestrów, który wynosi $DFF000. Adresy w tabeli zostały podane jako liczby szesnastkowe. Wszystkie rejestry możemy podzielić na cztery typy:

R (Read) — rejestr przeznaczony tylko do odczytu.
W (Write) — rejestr przeznaczony tylko do zapisu.
S (Strobe) — wpisanie danych do rejestru spowoduje jednorazowe zadziałanie układu.
ER (Early Read) — rejestr wyjściowy DMA, dostęp do niego ma tylko kontroler DMA.

Bezpośrednio po nazwie rejestru podaję w nawiasie jego adres względny (przesunięcie od adresu bazowego) oraz typ.

DMACONR (002) R
Odczyt sterownika DMA (ma do niego dostęp również AGNUS).

DSKDATR (008) ER
Odczyt danych z dysku do RAM.

ADKCONR (010) R
Odczyt sterownika dźwięku/dysku.

POT0DAT (012) R
Odczyt potencjometru w porcie 0.

POT1DAT (014) R
Odczyt potencjometru w porcie 1.

POTGOR (016) R
Odczyt danych portu potencjometru.

SERDATR (018) R
Odczyt portu szeregowego i statusu.

DSKBYTR (01A) R
Odczyt bajtu danych dyskowych i statusu.

INTENAR (01C) R
Odczyt zezwolenia przerwania (interrupt enable).

INTREQR (01E) R
Odczyt żądania przerwania (interrupt request).

DSKLEN (024) W
Długość bloku DMA dysku.

DSKDAT (026) W
Zapis danych z RAM na dysk.

SERDAT (030) W
Zapis danych (za pośrednictwem portu szeregowego) oraz bitów stopu.

SERPER (032) W
Sterowanie portem szeregowym i prędkością transmisji.

POTGO (034) W
Zapis danych portu potencjometru i bit startu.

STRHOR (03C) S
Sygnał synchronizacji poziomej (do tego rejestru ma również dostęp układu DENISE).

DMACON (096) W
Rejestr sterujący DMA (mają do niego dostęp układy: AGNUS, PAULA, DENISE).

INTENA (09A) W
Zezwolenie przerwania.

INTREQ (09C) W
Żądanie przerwania.

ADKCON (09E) W
Sterowanie dźwiękiem, dyskiem i UART.

Opisane poniżej rejestry i ich zastosowanie są identyczne dla czterech kanałów dźwiękowych; stąd też podam jedynie adresy rejestrów przydzielonych pozostałym kanałom.



AUD0LEN (0A4) W
Długość danych dźwiękowych kanału 0. Pozostałym kanałom przydzielono rejestry: AUD1LEN (0B4) — kanał 1, AUD2LEN (0C4) — kanał 2, AUD3LEN (0D4) — kanał 3.

AUD0PER (0A6) W
Okres próbkowania kanału 0. Pozostałym kanałom przydzielono rejestry: AUD1PER (0B6) — kanał 1, AUD2PER (0C6) — kanał 2, AUD3PER (0D6) — kanał 3.

AUD0VOL (0A8) W
Głośność kanału 0. Pozostałym kanałom przydzielono rejestry: AUD1VOL (0B8) — kanał 1 AUD2VOL (0C8) — kanał 2, AUD3VOL (0D8) — kanał 3.

AUD0DAT (0AA) W
Dane dźwiękowe kanału 0 (do przetwornika C/A). Pozostałym kanałom przydzielono rejestry: AUD1DAT (0BA) — kanał 1, AUD2DAT (0CA) — kanał 2, AUD3DAT (0DA) — kanał 3.

Rejestr DMACON służy do zamykania i otwierania poszczególnych kanałów DMA. Bity w tym rejestrze mogą być albo ustawione albo wyzerowane; decyduje o tym bit 15. Jeśli jest on w stanie logicznym 1 (ustawiony), to wszystkie ustawione bity słowa danych zostaną również ustawione. Jeżeli natomiast bit 15 jest wyzerowany, to wszystkie ustawione bity w słowie danych zostaną w DMACON wyzerowane.

Bit 15 (SET/CLR)	— decyduje o ustawieniu lub wyzerowaniu bitów
Bit 14 (BBUSY)	— blitter zajęty (tylko odczyt).
Bit 13 (BZERO	— wynik operacji blittera =0 (tylko odczyt).
Bity 12–11	— niewykorzystane.
Bit 10 (BLTPRI	— DMA blittera ma pierwszeństwo przed procesorem.
Bit 9 (DMAEN	— uaktywnienie ustawionych kanałów DMA (dla bitów 0–8).
Bit 8 (BPLEN)	— DMA dla płaszczyzn bitowych (bit planes).
Bit 7 (COPEN)	— DMA dla Coppera.
Bit 6 (BLTEN)	— DMA dla Blittera.
Bit 5 (SPREN)	— DMA dla duszków (sprites).
Bit 4 (DSKEN)	— DMA dla dysku.
Bity 3–0 (AUDxEN)	— DMA dla dźwięku dla kanału x (numer bitu odpowiada numerowi kanału)

Bit 9 (DMAEN) pełni rolę głównego przełącznika. Jeśli jest on wyzerowany, to wszystkie kanały DMA, bez względu na bity 0–8, będą nieaktywne. Jeżeli chcemy uaktywnić jakikolwiek kanał DMA, to oprócz ustawienia jego bitu (bity 0–8) musimy ustawić także bit 9 (DMAEN).

Zawartość rejestru DMACON możemy odczytać pod adresem $DFF002 (rejestr DMACONR). Stan bitu 13 i bitu 14 może poinformować nas czy blitter wykonuje jeszcze jakąś operację, oraz czy wynik wszystkich operacji blittera wyniósł 0.

Za pomocą PAULI możemy zarządzać poszczególnymi źródłami przerwań i generować sygnały przerwań dla procesora. Mamy do dyspozycji dwa rejestry: żądania przerwania (INTREQ) i maski przerwania (INTENA). Przeznaczenie bitów w obu rejestrach jest takie same. Poniżej podano kolejno: numer bitu, nazwę, adres autowektora, poziom oraz funkcję.

Bit 15 SET/CLR, —, —: Tak jak w DMACON.

Bit 14, INTEN, ($78), (6): Uaktywnienie przerwania.

Bit 13, EXTER, $78, 6: Przerwanie z CIA-B lub portu rozszerzeń.

Bit 12, DSKSYN, $74, 5: Rozpoznanie wartości synchronizacji dysku.

Bit 11, RBF, $74, 5: Bufor odbiorczy danych szeregowych zapełniony.

Bit 10, AUD3, $70, 4: Wyjście danych dźwiękowych kanału 3.

Bit 9, AUD2, $70, 4: Wyjście danych dźwiękowych kanału 2.

Bit 8, AUD1, $70, 4: Wyjście danych dźwiękowych kanału 1.

Bit 7, AUD0, $70, 4: Wyjście danych dźwiękowych kanału 0.

Bit 6, BLIT, $6C 3: Blitter gotowy.

Bit 5, VERTB, $6C, 3: Osiągnięto początek wygaszania pionowego.

Bit 4, COPER, $6C, 3: Przerwanie Coppera.

Bit 3, PORTS, $68, 2: Przerwanie z CIA-A lub portu rozszerzeń.

Bit 2, SOFT, $64, 1: Przerwania programowe.

Bit 1, DSKBLK, $64, 1: Wykonany transfer przez DMA dysku.

Bit 0, TBE, $64, 1: Bufor nadawczy danych szeregowych jest pusty.

Bit 14 (INTEN) w rejestrze INTENA pełni rolę głównego przełącznika. Przerwanie będzie wygenerowane jeśli bit INTEN i dwa odpowiednie bity w rejestrze INTENA i INTREQ zostaną ustawione. Przy ustawianiu bitów obowiązuje taka sama zasada jak w przypadku omawianego wcześniej rejestru DMACON.

Jeżeli chcemy wygenerować przerwanie wystarczy do rejestru INTREQ wpisać odpowiednią wartość (np. za pomocą rozkazu MOVE). Bit 14 w rejestrze INTREQ nie ma określonej funkcji, tak jak w INTENA, lecz jeżeli bit INTEN w INTENA jest ustawiony i ustawimy bit 14 w INTREQ, to wygenerowane zostanie przerwanie poziomu 6.

Należy pamiętać aby w rejestrze INTREQ wyzerować bit, który spowodował wytworzenie przerwania. Zawartość rejestrów INTREQ i INTENA możemy odczytać pod adresami $DFF01E (INTREQR) i $DFF01C (INTENAR).

BARTOSZ SMAGA

PRZEŁĄCZNIK PAL–NTSC dla AMIGI

Opisany przełącznik umożliwia przełączenie Amigi z pracy w systemie telewizyjnym PAL (stosowanym szeroko w Europie Zachodniej) na tryb pracy w systemie TV NTSC, używanym w USA, Kanadzie, Japonii i Filipinach. Dodam jeszcze, że NTSC USA i Kanady różnią się od NTSC stosowanego w Japonii.

Częstotliwość odchylania pionowego zostaje zwiększona do 60 Hz (w trybie PAL 50 Hz). Powoduje to, że migotanie obrazu jest nieco mniej widoczne. Ilość wyświetlanych linii zostaje zmniejszona z 625 do 525. Ponieważ nie ma róży bez kolców, również rozdzielczość pionowa zostaje zmniejszona z 256 (512) punktów na 200 (400). Jednak czasem można to uznać za zaletę: niektóre gry są tworzone właśnie dla systemu NTSC i w systemie PAL wyglądają jakby ekran był przesunięty nieco do góry.

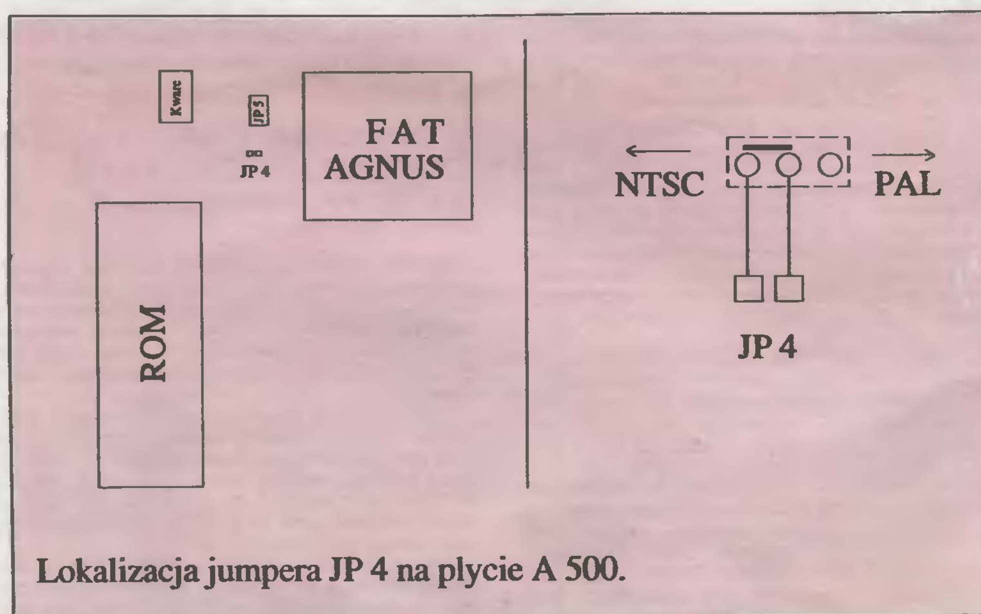
Teraz o konstrukcji: wszystko co musisz zrobić, to podłączyć wyłącznik do zworki JP4. Jego lokalizacja jest pokazana na rysunku.

Uwagi:

1. Agnus sprawdza stan jumpera tylko w momencie inicjalizacji systemu (reset), tak więc nie jest możliwe przełączenie systemu w trakcie pracy komputera.
2. Na niektórych monitorach i większości telewizorów wystąpią problemy z utrzymaniem synchronizacji pionowej obrazu.
3. Niektóre programy mogą działać nieco szybciej, gdyż przerwanie VBlank jest teraz wywoływane 60 razy na sekundę. Przykładem jest tu gra Lotus Turbo Challenge 2.

JERZY DUDEK

UWAGA!
DO WYKONANIA PRZERÓBKI UŻYWAJ TYLKO LUTOWNICY MAŁEJ MOCY, GDYŻ PRZEGRZANIE MOŻE SPOWODOWAĆ ODKLEJANIE SIĘ ŚCIEŻEK OBWODU DRUKOWANEGO. PONIEWAŻ ZWORKA (JUMPER) JP4 ZNAJDUJE SIĘ W BEZPOŚREDNIEJ BLISKOŚCI UKŁADU AGNUS NALEŻY ZACHOWAĆ MAKSYMALNĄ OSTROŻNOŚĆ. UKŁAD TEN JEST CZUŁY NA TEMPERATURĘ I NAPIĘCIA ELEKTROSTATYCZNE. JEŻELI NIE MASZ WPRAWY W POSŁUGIWANIU SIĘ LUTOWNICĄ, POWIERZ WYKONANIE PRZERÓBKI FACHOWCOWI. WYMIANA UKŁADU AGNUS WIĄŻE SIĘ Z WYDATKIEM RZĘDU 1.2 MILIONA ZŁOTYCH!!!



Lokalizacja jumpera JP 4 na płycie A 500.

AIRBUS A320 — KOLEJNY SYMULATOR

Lotnisko międzynarodowe we Frankfurcie nad Menem: Airbus A320 należący do Lufthansy kołuje do startu. Jeszcze dwie minuty i kapitan decyduje się startować. Ostatnie ruchy joysticka i na ekranie widać piękną panoramę pasa startowego. Napięcie wzrasta z każdą chwilą: czy długość pasa wystarczy na rozpęd? czy samolot nie waży zbyt dużo? czy do komputera pokładowego wprowadzone dobre dane odnośnie kursu? Kapitan. niepewny tego ostatniego, bierze głęboki oddech, naciska klawisz Pause i... zaczyna wprowadzanie danych na nowo.

Program firmy Thalion „Airbus A320" dla Amigi jest uznawany za jeden z bardziej realistycznych. Na przykład przebieg i charakterystyka lotu są (jak twierdzą wtajemniczeni) „prawie zgodne z oryginałem". Nic w

tym dziwnego. Każdy etap pracy nad programem kontrolowany był wnikliwie przez ekspertów z Lufthansy i konstruktorów A320. W celu zaznajomienia użytkownika z przeróżnymi aspektami lotu (programowanie pokładowego komputera, różne awarie, nawigacja itp.), do symulatora dołączono 250. stronicową książkę-przewodnik.

Gdy jako tako rozeznasz się już w nawigacji i sterowaniu A320 na poziomie „Trainings-Mode", przechodzisz do „Duty-Mode", w którym stajesz się powoli prawdziwym pilotem, przeobrażając się stopniowo z pilota najniższego rangą („Student-Pilot") w starego wyjadacza powietrznego („Chief-Pilot"). Tu uwaga: matka natura i technika lubią płatać czasem figle, przyprawiające o zawrót głowy. I tak w Airbusie nawalają często instrumenty nawigacyjne, części mechaniczne, a niesprzyjające warunki atmosferyczne wymuszają ciągłą obserwację przyrządów i wprowadzanie korekt do komputera. Na końcu każdej symulacji otrzymujesz adekwatną do uzyskanych wyników notę, która może być zapamiętana na dyskietce. Jeśli w swych zmaganiach zdołałeś uzyskać rangę Chief-Pilot, warto ową dyskietkę wysłać do firmy Thalion, producenta programu. Otrzymasz wtedy „Złotą odznakę Airbus".

A dla tych, którzy chcą naprawdę poważnie „popracować" nad symulatorem A320, przewidziano jeszcze jeden konkurs. Najlepszych dziesięciu „pilotów" wzięło udział w wielkim finale, który odbył się na wielkiej wystawie lotniczej w Berlinie, w dniach 15–21 czerwca 1992. Zwycięzcy (trzy pierwsze miejsca) „polatali" w prawdziwym, profesjonalnym symulatorze lotu A320, w którym normalnie szkolą się piloci Lufthansy.

Artykuł ten ma jedynie zasygnalizować, że wśród powodzi wojskowych symulatorów pojawił się nareszcie program symulujący samolot pasażerski. O ile mi wiadomo „A320" dotarł już do rąk Sidewindera i myślę, że niebawem będziemy mogli podać Czytelnikom bliższe szczegóły na temat tego programu.

Na podstawie „COMM"
Opracował CHRISTIAN GRZENKOWICZ

RAPORT

EPROM z POLSKĄ CZCIONKĄ dla DRUKAREK STAR LC-10/20

Nasz język ojczysty ma pewną cechę, która utrudnia pracę z komputerem. Jest nią obecność polskich znaków diakrytycznych (czyli ą, ę, ó, ł itd.). Niestety zachodni producenci sprzętu i oprogramowania nie ułatwiają nam życia pod tym względem. Jak się okazuje, możemy jednak liczyć na rodzimą myśl techniczną.

Takie stwierdzenie nasunęło się po przetestowaniu drukarki Star LC-20, pracującej z zainstalowaną pamięcią EPROM zawierającą polskie znaki. Aby stać się szczęśliwym posiadaczem takiego urządzenia należy dostarczyć do serwisu komputerowego firmy Amiga S.C. (dawny Klub Komputerowy „Stodoła") nieco przyciężką drukarkę lub (na własne ryzyko) sam EPROM z niej wyjęty. Usługa świadczona przez serwis polega na przeprogramowaniu standardowego zestawu znaków na zestaw zgodny z przyjętym w Polsce standardem Mazovii. Oprócz tego użytkownik otrzymuje dyskietkę zawierającą pliki z informacjami w jaki sposób należy przystosować do tak przerobionej drukarki popularny wśród posiadaczy Amigi edytor Cygnus. Dodatkowo dyskietka zawiera również specjalne pliki sterujące niezbędne do poprawnej pracy edytora z tak zmienioną EPROM. Istnieje możliwość przeróbki Cygnusa na miejscu, pod okiem fachowców, pod warunkiem jednak, że klient przyniesie własną kopię tego edytora na własnej dyskietce.

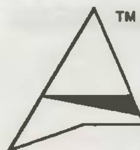
Opisywane tu rozwiązanie ma sporo zalet w porównaniu z programową przeróbką znaków. Po pierwsze: znaki zainstalowane są w pamięci stałej, przez co nie zajmują pamięci RAM drukarki i dzięki temu bufor jest znacznie większy. Przy pracy w trybie „download" znaki zajmują sporo pamięci RAM w drukarce co zdecydowanie spowalnia transmisję (mały bufor dla danych). Po drugie: nie spotkałem żadnego edytora, który w trybie download przerabia wszystkie kroje czcionek. Polscrible drukuje tylko w trybie Draft, a Transcript przeprogramowuje tylko jeden krój NLQ — Courier. Po trzecie: kody znaków wpisanych do EPROM są zgodne ze standardem Mazovia; takie rozwiązanie ułatwia zdecydowanie wykorzystanie drukarki tak z Amigą jak i z pecetem.

W przypadku współpracy z Amigą (bo przerobiona w ten sposób drukarka może być wykorzystana z dowolnym pecetem pracującym w standardzie Mazovii) można również wspomnieć o pewnych niedogodnościach. Lepiej byłoby, gdyby kody były zgodne np. z kodami Cygnusa. Wtedy większość oprogramowania Amigi drukowałaby po polsku bez przeróbek. Niezbyt trafnym rozwiązaniem wydaje mi się akurat wybór edytora Cygnus, który może jest najlepszy jako edytor, lecz opcje drukowania są w nim ograniczone do minimum. Nie ma możliwości ustawiania kroju czcionki, marginesów, rozmiaru papieru, gęstości drukowania itp. Lepszy pod tym względem jest np. Transcript, który ma znacznie bardziej rozbudowaną część drukującą i daje użytkownikowi wszystkie wyżej wymienione opcje. Zaawansowany użytkownik nie powinien mieć problemów z przeróbką innych edytorów.

Pomimo drobnych wad (chyba nic nie jest od nich wolne) przeróbka jest pomysłowym ominięciem problemów związanych z drukowaniem po polsku, warte polecenia użytkownikom tak Amig jak i pecetów. Jeżeli do dużej użyteczności samego rozwiązania dodamy jeszcze niewielką cenę związaną z wykonaniem w/w usługi, to w efekcie otrzymamy atrakcyjny produkt, który powinien zadowolić nawet najbardziej wymagających użytkowników.

JERZY DUDEK

WYKONAWCA:
AMIGA S.C., ul. Batorego 10, Warszawa, tel. 25-60-31 wew. 35.



S.C. Alderan

Alderan S.C.
Korotyńskiego 19a/55, 02-123 Warszawa
telefony: 659-18-21
oraz 44-74-82...7 w. 221
fax: 628-04-10

S.C. Alderan prezentuje Państwu

Zestaw Biznesowy

- swój najnowszy pakiet programów, przeznaczonych dla osób prowadzących działalność gospodarczą. Programy zawarte w pakiecie służą do:

- prowadzenia Podatkowej Księgi Przychodów i Rozchodów,
- prowadzenia kadr,
- prowadzenia działu płacowego,
- prowadzenia magazynu,
- prowadzenia ewidencji środków trwałych,
- dokonywania odpisów amortyzacyjnych od środków trwałych,
- sporządzania kosztorysów

Programy zawarte w pakiecie są proste w obsłudze, co powoduje, że mogą one być obsługiwane przez osoby nie znające się na komputerach - wymagają one jednak pewnej znajomości odpowiednich dziedzin księgowości. Programy są oczywiście zgodne z najnowszymi ustawami i rozporządzeniami Ministra Finansów. Ponadto ich elastyczna konstrukacja umożliwia użytkownikowi wprowadzanie zmian, na przykład stawek opodatkowania.

cena: 1.990 tys. zł

Wymagania sprzętowe pakietu: podstawowa konfiguracja Amigi

Prowadzimy detaliczną sprzedaż wysyłkową (odbiorca płaci przy odbiorze, do ceny doliczamy koszty ponoszone na rzecz Poczty). Każdy, kto zamówi tą drogą więcej niż trzy tytuły, otrzyma gratis Anty-Virus, a powyżej czterech - program Notes.

Do wszystkich oferowanych programów posiadamy wszelkie prawa i jesteśmy ich jedynym legalnym dystrybutorem.

Poszukujemy poważnych firm chcących zostać naszymi dealerami. Oferujemy korzystne warunki.

Zainteresowani jesteśmy również w nawiązaniu kontaktów ze zdolnymi programistami, a także grafikami, szczególnie specjalizującymi się w animacji trójwymiarowej.

Poza Zestawem Biznesowym oferujemy także najwyższej jakości programy użytkowe:

* **WordTeacher 2.0** - najnowsza wersja dobrze znanego i wypróbowanego programu do nauki języka angielskiego (pisowni i wymowy). Ma wbudowane dwa pełnosprawne słowniki: polsko-angielski i angielsko-polski (35 tysięcy słów). WT 2.0 wykorzystuje syntetyzer mowy, co umożliwia maksymalnie wierne odtwarzanie wymowy angielskiej. Dzięki metodom nauki zastosowanym w tym programie możliwe jest opanowanie z jego pomocą nawet 170 słów w ciągu godziny!
[komputer: Amiga, cena 88.000 zł]

* **A-Word** - pierwszy słownik angielsko-polski z prawdziwego zdarzenia, przeznaczony dla komputerów Amiga, bijący konkurencję na głowę zarówno liczbą słów, jak i wykonaniem. Superszybki (napisany w 100% w języku maszynowym), w pełni wykorzystujący wielozadaniowość Amigi. W zależności od upodobań i ilości dostępnej pamięci, możemy uruchomić go "na oknie" lub całym ekranie, możemy też zamknąć jego okno lub ekran, zostawiając go w pamięci jako program "drzemiący", który możemy w każdej chwili uaktywnić kombinacją klawiszy, gdy natrafimy podczas pracy na nieznane słowo. Wśród haseł uwzględniono także wszelkie terminy anglojęzyczne związane z Amigą!
[komputer: Amiga, cena 120.000 zł]

* **Twój Pierwszy Angielski** - wspaniała nauka języka angielskiego dla dzieci. Na program składa się 11 scen, w których dziecko ma za zadanie rozpoznać m. in. owoce, kolory i liczby. Nauka odbywa się z wykorzystaniem animacji komputerowej i syntezy mowy, zaś na końcu uczeń przystępuje do egzaminu. Twój Pierwszy Angielski jest programem najwyższej jakości, czego dowodem może być jego test w czwartym numerze C&A.
[komputery: Amiga/Commodore 64 (dysk i kaseta), cena 180.000 zł]

* **Ortografia (Gra Słów)** - zestaw czterech gier rozwijających wyobraźnię, spostrzegawczość, a przede wszystkim wiedzę z dziedziny ortografii (wbudowany słownik zawiera 10.000 słów prosto ze Słownika Ortograficznego). Program został skonstruowany z uwzględnieniem wszelkich reguł nauczania, nie jest, na przykład, możliwe uzyskanie na ekranie wyrazu błędnie napisanego - w pamięci utrwala się tylko poprawna pisownia.
[komputer: Amiga, cena 68.000 zł]

* **Geometria Konstrukcyjna** - wszystko o geometrii, nauka wszelkich twierdzeń, sposobów rozwiązywania zadań, wszystko w oparciu o przykłady, z wykorzystaniem animacji komputerowej. To trzeba mieć!
[komputer: Amiga, cena 79.000 zł]

* **Emulator 1.3** - rewelacja. Za ułamek ceny przeróbki hardware'owej mogą Państwo sprawić, że Wasza Amiga 500 Plus będzie w pełni kompatybilna z poprzednimi modelami. Dzięki emulacji systemu operacyjnego w wersji 1.3, znikną wszystkie Wasze kłopoty z uruchamianiem programów. Zgodność programowa Amigi Plus z uruchomionym Emulatorem1.3 jest stuprocentowa dzięki temu, że program całkowicie wyłącza system operacyjny w wersji 2.0 i zastępuje go pełnosprawnym Kickstart'em 1.3!
Uwaga: najnowsza wersja Emulatora1.3 pracuje poprawnie również z najnowszymi Amigami 600 i 300!
[komputer: Amiga 500+/2000+, cena 99.000 zł]

* **Pierwsze Kroki** - program zawierający kilkadziesiąt połączonych z tekstami rysunków, wyjaśniających obsługę Amigi, oprogramowania systemowego oraz sposób podłączania urządzeń zewnętrznych. Zawiera też wyjaśnienia kilkudziesięciu terminów związanych z Amigą. Idealny dla początkujących Amigowców, jak również dla firm sprzedających Amigi (zapewniamy również nalepki na pudełka).
[komputer: Amiga, cena 48.000 zł]

* **Chemia 2.0** - najnowsza, znacznie rozbudowana w stosunku do poprzedniej, wersja programu zawierającego wiadomości z zakresu chemii nieorganicznej. Znajdą tu Państwo w formie graficznej wszelkie informacje z tablicy Mendelejewa, jak również wiadomości z dziedziny mechaniki kwantowej. Program umożliwia automatyczne wyszukiwanie wszelkich zależności i podobieństw grup pierwiastków.
[komputer: Amiga, cena 59.000 zł]

* **Biorytmy 2.0** - program ten powie Państwu wszystko o Waszej kondycji psychicznej, fizycznej oraz intelektualnej.
[komputer: Amiga, cena 42.000 zł]

* **Notes** - bardzo wygodna, elastyczna w swej konstrukcji i prosta w obsłudze podręczna baza danych.
[komputer: Amiga, cena 200.000 zł]

* **Piórko** - prosty w obsłudze, ale o dużych możliwościach, dedykowany głównie dzieciom program graficzny.
[komputer: Amiga, cena 59.000 zł]

* **Zestaw Antywirusowy** - kompilacja najnowszych wersji najlepszych programów antywirusowych public domain, jakie dotychczas napisano.
[komputer: Amiga, cena 42.000 zł]

AZTEC C

Programiści jako ludzie z natury leniwi, zawsze starali się ułatwić sobie pracę. Gdy mieli już dość wpisywania w pierwsze komputery tysięcy zer i jedynek, wymyślili języki programowania. Najpierw były to asemblery, potem proste języki wysokiego poziomu, aż do języków strukturalnych. Jednak mimo nowych, coraz lepszych języków, systemy operacyjne oraz programy w pełni wykorzystujące możliwości komputera, nadal pisano w asemblerze. W ramach dalszego ułatwiania sobie życia, stworzono język „C". Łączy on łatwość programowania języków wysokiego poziomu z efektywnością asemblera, wypierając go ostatnio z obszarów, w których był dotąd tradycyjnie stosowany. Wiąże się to również ze spadkiem cen pamięci i wzrostem prędkości komputerów (program maszynowy jest zazwyczaj krótszy i szybszy).

Sytuacja ta zaowocowała systemem operacyjnym Amigi, napisanym w języku BCPL, będącym przodkiem języka „C". Nie powinno więc nikogo dziwić, że „C" jest świetnie dopasowany do współpracy z systemem, pozwalając w pełni wykorzystywać możliwości naszego komputera.

Jednym z lepszych kompilatorów tego języka jest AZTEC „C" firmy Manx. Pierwsza wersja tego kompilatora dla Amigi pojawiła się już w roku 1987, czyli praktycznie na początku kariery tego komputera. Była to wersja 3.4 a, do dziś najczęściej spotykana w naszej strefie klimatycznej ze względu na małe wymagania sprzętowe (wystarczy jedna stacja dysków i 512 KB pamięci). Niestety komfort obsługi — dotyczy to również nowszych wersji — jest daleki (żeby nie powiedzieć bardzo daleki) od tego co dają produkty firmy Borland na komputerach klasy PC.

Aztec „C" pracuje w trybie wsadowym. Droga do tekstu źródłowego — który możemy napisać przy pomocy dowolnego edytora pozwalającego na zapis w czystym kodzie ASCII — do programu wynikowego, prowadzi przez kompilator (cc), asembler (as) i linker (ln), zwany od niedawna w Polsce konsolidatorem. W międzyczasie zapewne trzeba będzie uporać się z błędami, które kompilator sygnalizuje w bardzo oszczędny sposób, podając tylko ich numery i opis w dwu, trzech słowach.

Wszystkie trzy elementy kompilatora wywoływane są z poziomu AmigaDos (CLI lub Shell) z podaniem nazwy kompilowanego programu i opcji. Kompilator w pełni respektuje standard języka wraz ze standardowymi funkcjami. Oczywiście nie ogranicza się tylko do tego, ma on np. szereg rozszerzeń takich jak dodatkowe funkcje, czy możliwość włączania do tekstu programu procedur w asemblerze. Kompilator daje również dostęp do wszystkich funkcji bibliotek systemowych Amigi, tak jakby to były normalne funkcje języka; wystarczy w tym celu podać nazwę i parametry. W tym celu w katalogu INCLUDE umieszczone są pliki z definicjami zmiennych i struktur systemu operacyjnego. Jeśli chcemy z nich skorzystać, to należy wstawić te, które są nam potrzebne do tekstu programu w postaci:

```
#include <nazwa pliku.h>
```

Szczególnie dotyczy to pliku function.h, bez którego niemożliwe jest użycie funkcji systemowych.

Program jest tłumaczony przez kompilator do postaci tekstu w języku asemblera; kod ten następnie trzeba skompilować asemblerem znajdującym się w zestawie. Jeżeli nie zażyczymy sobie inaczej, ten ostatni etap zostanie wykonany automatycznie. Kompilator asemblera można oczywiście zastosować do kompilacji napisanych przez nas programów asemblerowych.

Ostatnim etapem jest połączenie niezależnie kompilowanych fragmentów programu i ewentualnych procedur maszynowych, z bibliotekami funkcji języka „C" w gotowy program wykonywalny. Temu celowi służy linker. Jest on na tyle „inteligentny", że do gotowego programu dołącza z bibliotek jedynie te funkcje, które zostały wykorzystane. Jest to bardzo cenna zaleta, gdyż np. konieczna do działania programu biblioteka c.lib zajmuje ponad 100 KB. Tyle miałby program który nic nie robi, a tymczasem mieści się on w niecałych dwóch kilobajtach (znajduje się w nich między innymi procedura otwarcia biblioteki systemowej dos.library).

Standardowe procedury wejścia i wyjścia komunikują się z oknem CLI, jeśli z niego program został uruchomiony. Na wypadek wczytywania z poziomu Workbencha, do programu dołączana jest procedura, która otwiera niezbędne okno, ale tylko wtedy, gdy w TOOL TYPES ikony naszego programu znajduje się informacja o jego wielkości (WINDOW=CON:x/y/ szer. /wys./ Tytuł).

Nowsze wersje kompilatora zawierają dodatkowo szereg programów ułatwiających programowanie. Jest to między innymi debugger, pozwalający śledzić wykonanie programu z poziomu tekstu źródłowego, co znakomicie ułatwia wyszukiwanie błędów.

Najstarsza wersja kompilatora nie może poszczycić się zbyt dużą prędkością kompilacji, w najnowszej 5.0a, prędkość ta jest o wiele większa. Szczególnie dużo czasu zajmuje komunikacja z dyskiem, z którego kompilator korzysta bardzo obficie. Kompilacja odbywa się na dysku, odczytywane są dołączane pliki „include" i biblioteki dla linkera. Z własnej praktyki wiem, że wszystkie potrzebne elementy warto umieścić w RAM-dysku, co znacznie przyspiesza działanie. Oczywiście sytuację rozwiązuje również dysk twardy.

Prędkość programów wynikowych jest kilkadziesiąt do kilkuset razy większa od szybkości programów w BASIC i ustępuje ona jedynie asemblerowi. Dzięki możliwości włączania kodu maszynowego w tekst programu, bardziej czasochłonne fragmenty programu można przyspieszyć.

Mimo wielu niewygód omawiany kompilator jest potężnym narzędziem. Większość programów użytkowych powstało na Amigę właśnie w „C", więc jeśli jeszcze nie znasz tego języka, warto pomyśleć o jego poznaniu. Najlepszym bardzo przystępnym podręcznikiem jest książka jego twórcy, Denisa Ritchiego oraz Briana Kernighana „Język C" (wydana przez WNT w 1988 roku ISBN 83-204-1067-3). Kurs tego coraz bardziej popularnego języka publikowany był także w Bajtku.

Oczywiście ze zrozumiałych względów niniejszy opis w żadnym wypadku nie jest w stanie wyczerpać tematu; ewentualne wątpliwości postaram się wyjaśnić listownie za pośrednictwem redakcji.

MARCIN JAZGAR

NEGATRON



NEGATRON jest jedną z nielicznych polskich gier komputerowych dla Commodore 64. Grę tę można zaliczyć do bardzo ostatnio popularnych gier logicznych. Zadaniem gracza jest odwrócenie w określonym czasie wszystkich widocznych na ekranie kwadracików na drugą stronę (wersja, jaką otrzymaliśmy od dystrybutora była zapisana na kasecie).

Nie jest to jednak takie proste, gdyż przy odwracaniu kolejnych kwadratów jednocześnie odwracane są także sąsiednie. O liczbie brakujących do sukcesu kwadracików oraz o czasie jaki pozostał nam do końca gry informuje nas komputer. Pomysł gry nie jest nowy; jest to jedna z wersji znanej gry „odwracanka"...

Dobra grafika i dźwięk sprawiają, że w NEGATRON można grać przez wiele godzin — jest to zabawa zarówno dla dzieci, jak i dla dorosłych. Ciekawie została opracowana opcja dla dwóch osób grających równocześnie.

Na koniec tego krótkiego opisu wypada pogratulować autorom napisania tak całkiem dobrego jak na polskie warunki programu. I jeszcze chciałbym dodać, że mój osobisty rekord to brak jednego kwadracika do zakończenia gry.

MARIUSZ FERDYN

DYSTRYBUTOR:

Fundacja Edukacji Technologicznej, ul. Burdzińskiego 5, Warszawa, tel. 18-01-76.

- Producent: **Fundacja Edukacji Technologicznej**
- Rodzaj gry: logiczna
- Komputer: **Commodore 64 i Commodore 128 w trybie C-64**
- Wymagania: —

BLUE ANGEL 69

Do tej gry nie potrzeba żadnej wymyślnej historii z pogranicza sf i fantasy. Ot po prostu — wyobraź sobie planszę 8x8, na której jakiś maniak rozsypał cyfry od 1 do 11 (kolor fioletowy — dodatnie, kolor czerwony — ujemne). To jest pole bitwy. Gracz porusza się w poziomie, komputer w pionie. Cel jest tylko jeden — wygrać większą ilością punktów. Niby proste, co? Nieprawda! Kombinacji jest tyle, że możemy zaplanować tylko kilka posunięć naprzód. Należy też pamiętać, że nie zawsze upragnione +11 da nam przewagę. Wydawałoby się, że jest to gra niegodna uwagi. No, ale dlaczego w takim razie dostarczyła mi zabawy na kilka dni.

Istnieje także możliwość grania na czas lub ukrycia liczb, których nie ma w linii poziomej, po której porusza się kursor — czyni to grę dużo trudniejszą, ale przy okazji o wiele zabawniejszą (szczególnie dla 2 graczy).

Tak. Czasami prosty pomysł może być dobrym pomysłem. I tak jest w wypadku Blue Angel. Jest to naprawdę świetna gra logiczna.

Muzyka całkiem przyjemna dla ucha, ale lepiej nie włączać fx, bo gdy robot, z którym toczysz pojedynek, zacznie ponaglać swoim „COME ON!", to można zagubić się w wymyślonej przed chwilą strategii. Grafikę stanowi szereg zdjęć połączonych z „normalną" grafiką, ogólnie całkiem przyjemny widok.

BAD

DYSTRYBUTOR:
IPS Computer Group, Warszawa.

- Producent: **Eurogold**
- Rodzaj gry: logiczna
- Komputer: **C-64, Amiga, IBM, Atari ST**

HARPOON

„UWAGA!!! Z lewej burty zbliżają się dwie torpedy!!! Eskadra wrogich samolotów nadlatuje z północy!!! Grupa patrolowa ZZ1 została zestrzelona!!!"

Tak zwykle wygląda pierwszy kontakt z grą „Harpoon": nie wiemy co się dzieje, z każdej strony do nas strzelają, kolejne okręty z naszego konwoju idą na dno. „Harpoon" jest grą symulacyjną uznawaną przez znawców tematu, w chwili obecnej, za najlepszą w swojej klasie.

Twórcy programu postarali się o wszystko; bogactwo podawanych informacji jest niezmierzone. O każdej jednostce nawodnej, podwodnej czy też latającej dostajemy wyczerpujące wiadomości: jej wygląd zewnętrzny, dane techniczne, uzbrojenie, odległość skutecznego rażenia poszczególnych rodzajów roni, zasięg radarów itp. W grze możemy się wcielić w jedną z dwóch postaci — w naczelnego dowódcę sił morskich Stanów Zjednoczonych lub naczelnego dowódcę sił morskich imperium radzieckiego. Naszym zadaniem jest wygrać wojnę, obojętnie, po której ze stron byś się nie znalazł. Możesz prowadzić wojnę nie tylko na morzu, lecz i w powietrzu, wysyłając samoloty z lotniskowców lub przybrzeżnych lotnisk.

Na samym początku gry należy pamiętać o uaktywnieniu radarów, które pozwolą nam na w miarę szybkie zlokalizowanie jednostek wroga i podjęcie odpowiednich środków bezpieczeństwa. Możemy także wysłać patrol helikopterów w celu przeczesania pobliskich obszarów wodnych w poszukiwaniu ukrytych w głębinach łodzi podwodnych. W grze kierujemy całymi konwojami, nadajemy im prędkość i okeślamy dla nich odpowiedni kurs. Możemy także zmienić szyk statków w poszczególnych konwojach. Programiści nie zapomnieli nawet o takim szczególe jak szalejące sztormy morskie, będące w stanie zniszczyć patrol helikopterów.

Dla ułatwienia możesz wprowadzić na mapy sytuacyjne siatkę kartograficzną, jest także możliwość zamiany symboli obrazkowych (konwoje, eskadry samolotów itd.) na prawdziwe symbole sztabowe, które czynią grę niemal realną. Do gry dołączono dwie dyskietki ze scenariuszami: na pierwszej znajduje się 13 propozycji wydarzeń, które mogłyby być konsekwencją współczesnej światowej polityki, na drugiej znalazło się 16 wariantów ogólnego tematu, jakim jest obrona wybrzeży Północnej Ameryki. Dostępne są również różne dodatki do gry, takie jak np. edytor scenariuszy pozwalający na układanie i edycję wymyślonych przez grającego misji.

Do głównych mankamentów gry „Harpoon" zaliczyłbym przede wszystkim jej małą szybkość. Zdarza się, że trzeba bardzo długo czekać na wykonanie wybranej przez nas opcji, co sprawia, że gra traci na swojej atrakcyjności. Choć jest to program stosunkowo skomplikowany nie należy się zniechęcać — Ci którzy go poznają na pewno nie pożałują.

BAS



RAPORT

MAX PL

Fundacja Edukacji Technologicznej nie od dziś znana jest z własnych opracowań — wymienię tu jedynie WARSAW BASIC czy np. PRINT PL. W celu zabezpieczenia się przed kradzieżą programy są zapisywane w pamięciach EPROM co czyni je niedostępnymi dla większości handlarzy. Dziś chciałbym przedstawić najnowszy produkt tej firmy — MAX PL.

KONSTRUKCJA

MAX PL jest standardowym modułem przyłączanym do portu MEMORY EXPANSION (lub EXPANSION) Commodore 64 i 128. W testowanym egzemplarzu obudowa modułu uniemożliwiała włożenie karty odwrotnie (jedna z listew blokujących jest grubsza). Moduł wyposażony jest w przycisk RESET ukryty wewnątrz obudowy co zapobiega przed jego przypadkowym wciśnięciem. Aby wykonać RESET trzeba będzie posłużyć się długopisem lub np. zapałką.

Na karcie zainstalowano pamięć EPROM o pojemności 64 KB; dzięki temu moduł ten ma dość bogate możliwości.

GWARANCJA I SERWIS

MAX PL jest objęty pełną gwarancją przez 12 miesięcy. Gwarancja obejmuje jedynie moduły nie noszące na sobie śladów przeróbek, „udoskonaleń" i prób otwarcia. Dodatkowo producent zapewnia wszystkim nabywcom doradztwo w zakresie wykorzystania tej karty.

INSTRUKCJA OBSŁUGI

Jak na razie stanowi ją jedna kartka formatu A4, i jest to naszym zdaniem zbyt mało (przy większym formacie możnaby podać np. programy przykładowe demonstrujące zastosowanie nowych poleceń BASIC). Zgodnie z obietnicą producenta problem ten przestanie istnieć zaraz po uruchomieniu własnego stanowiska DTP, które jak dotąd stanowiło wąskie gardło firmy. Tym niemniej trzeba przyznać, że sama instrukcja jest napisana treściwie i bez niedomówień, choć niektóre zagadnienia są opisane nieco za skąpo.

ZAWARTOŚĆ EPROM

Bezpośrednio po włączeniu komputera z dołączonym modułem dostępny jest tryb standardowy (BASIC V2 i 38911 bajtów pamięci RAM). Karta może również pracować w trybie PL; w trybie tym użytkownik ma do dyspozycji 39935 bajtów RAM oraz polskie znaki na klawiaturze i ekranie. Uzyskiwanie polskich liter zostało zrealizowane w oparciu o wygodny standard przyjęty z edytora TEXTOMAT — klawisz C= i litera (np. C= + a daje „ą" itp.).

W trybie standardowym nie zachowano kolorów ekranu zmieniając je na kolor czarny. Po naciśnięciu dwóch klawiszy przechodzisz do dość obszernego MENU, z którego możesz wybrać interesujący Cię program. A oto dostępne programy:

Obsługa magnetofonu

1) Program TURBO do magnetofonu (zgodny z TURBO ROM).
2) Program Turbo Save/Load.



3) Program kopiujący z dyskietek na kasety.
4) Program do regulacji skosu głowicy.

Obsługa stacji dysków

1) Program TURBO przyspieszający ośmiokrotnie wczytywanie z dyskietek.
2) Obsługa stacji dysków za pomocą poleceń DOS WEDGE 5.0.

Rozszerzenie BASIC V2.0

Użytkownik otrzymuje zestaw 9 dodatkowych poleceń dzięki którym dostępne jest programowanie proceduralne. Są to:

£m (MEMORY)
Wyświetla informacje o liczbie wolnych bajtów na zmienne i program oraz informację o numerze poziomu zagnieżdżenia procedur.

£n (NEW)
Zakłada nowy poziom procedur wykonując NEW na poziomie o jeden wyższym od tego na którym instrukcja ta została wydana.

£p (PROCEDURE)
Pozwala na przejście o jeden poziom wyżej od tego na którym polecenie to zostało wydane.

£o (OFF)
Powoduje przejście o jeden poziom niżej (przeciwieństwo polecenia £p).

£c (CALL)
„nazwa", nr urządzenia, lista parametrów (CALL)
Pozwala na wywołanie procedury z pamięci zewnętrznej i przekazuje jej ze swojej listy wartości parametrów. Wywołana procedura korzysta ze zmiennych lokalnych.

£h (HEADER)
„nazwa", lista parametrów formalnych (HEADER)
Nagłówek procedury przechowywanej w pamięci zewnętrznej, której nazwa i lista parametrów ma być zgodna z nazwą i listą parametrów w wywołaniu.

£e (END)
Pozwala na powrót z podprogramu wywołanego za pomocą CALL.

£a"nazwa", nr urządzenia (APPEND)
Wywołuje podprogram korzystający ze zmiennych globalnych to jest tych samych co wywołujący go program.

£f (FINISH)
Pozwala na powrót z podprogramu wywołanego za pomocą APPEND.

Dodatkowo, MAX PL jest również wyposażony w szereg poleceń przeznaczonych do obsługi stacji dysków. Dzięki nim w łatwy sposób możesz sformatować dyskietkę, zmienić nazwę pliku, skasować niepotrzebne zbiory czy wykonać kopię ważniejszych plików na tej samej dyskietce. Nie zabrakło oczywiście instrukcji pozwalającej na odczyt katalogu dyskietki i kanału rozkazowego stacji, możliwa też jest inicjalizacja (reset) stacji.

Gry komputerowe

1) NEGATRON (logiczna)
2) CHEESEBURGER (zręcznościowa)
3) GRIDRUNNER (zręcznościowa)
4) BILLARD (bilard)

NASZ WERDYKT

Jest to już kolejna karta Fundacji Edukacji Technologicznej stanowiąca przedłużenie linii modułu HELP PL. Naszym zdaniem MAX PL jest kolejną ciekawą propozycją zawierającą kilka unikalnych opcji (m.in. programowanie proceduralne i polskie znaki).

Z drugiej strony zastanawiamy się, czy istniała konieczność wpisywania do EPROM aż czterech gier komputerowych; gdyby w miejscu jednej z nich ulokować monitor języka maszynowego lub nawet asembler, atrakcyjność karty z pewnością by wzrosła. Znając jednak operatywność duetu Radziszewski — Gajewski wierzymy, że niebawem zobaczymy taką kartę w ofercie Fundacji; zanim to jednak nastąpi polecamy zapoznanie się z MAX PL, gdyż jest to moduł na pewno tego wart.

PIOTR LISZEWSKI
KLAUDIUSZ DYBOWSKI

ZALETY:
+ prostota obsługi,
+ wbudowane polskie znaki
+ wbudowane polecenia umożliwiające programowanie proceduralne
+ duża liczba wbudowanych programów użytkowych
+ przycisk RESET umieszczony tak, aby nie spowodować przypadkowego skasowania pamięci komputera.

WADY:
— skąpa instrukcja,
— zbyt dużo gier.

NAJLEPSZE z NAJLEPSZYCH

edytory tekstu dla Commodore 64/128

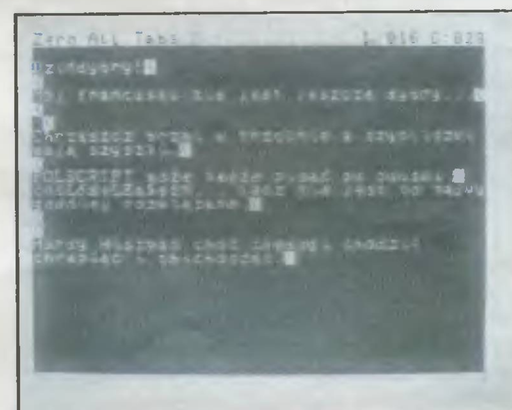
Choć komputerami zajmuję się od 1984 roku, to jednak przez dość długi okres czasu „obwąchiwałem" edytory tekstu raczej nieufnie. Powodów było kilka: po pierwsze programów z polskimi znakami było wtedy jak na lekarstwo, po drugie nie miałem stałego dostępu do horrendalnie wówczas drogiej drukarki.

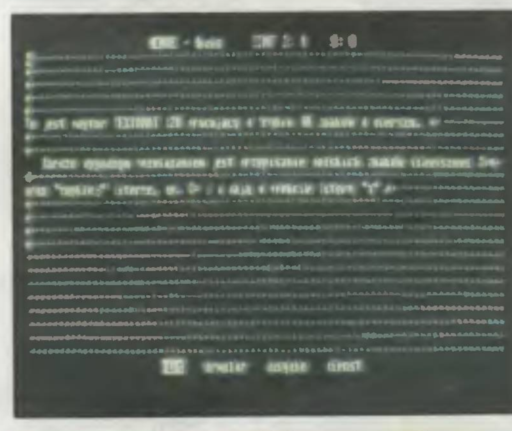
Do zastosowania edytora tekstu zmusiły mnie właściwie warunki zewnętrzne. Najpierw zaczęło się od prostych edytorków typu SKYWRITER a skończyłem na TEXTOMACIE 128. W międzyczasie, do szczególnie „ważnych" prac, wymagających wydruków wysokiej jakości stosowałem Fontmaster 64 II i Fontmaster 128. Po drodze, niejako mimochodem, miałem okazję zapoznać się z programami typu Polscript czy Star Texter.

Zanim przejdziemy do charakterystyki najlepszych moim zdaniem programów chciałbym na chwilę zatrzymać się nad wymaganiami jakie należałoby stawiać edytorom tekstu.

1. Twój edytor powinien mieć możliwość ustawiania wszystkich marginesów (górnego, dolnego, prawego i lewego. Formatowanie tekstu na ekranie zgodnie z ustawionymi przez Ciebie marginesami jest pożyteczne ale niekonieczne.
2. Im więcej dostępnych krojów czcionki, tym lepiej dla Ciebie. Jeśli często posługujesz się niestandardowymi znakami warto rozejrzeć się za programem wyposażonym w edytor czcionek. Umożliwi Ci to oprogramowanie olbrzymiej większości Twoich potrzeb. Dobrze byłoby również, aby Twój edytor pozwalał na korzystanie z wielu krojów pisma jednocześnie.
3. W dobrym edytorze wszelkie zmiany stylu pisma są widoczne na ekranie (podkreślenia, tryby potęgowy i indeksowy, wytłuszczenia, kursywa itp). Jeśli nie widać zmian w trakcie edycji być może będą one widoczne podczas przeglądu dokumentu przed wydrukiem (tzw. opcja PREVIEW).
4. Dobry edytor umożliwi Ci obejrzenie sformatowanego tekstu w takiej postaci w jakiej pojawi się on na papierze.
5. Jeżeli piszesz do firmy czy instytucji wymagającej zachowania określonych standardów (np. 30 wierszy po 60 znaków w każdym, lewy margines o szerokości 5 cm, prawy o szerokości 1 cm), to byłoby dobrze nabyć taki program, który pozwala Ci na PŁYNNĄ regulację odstępu międzywierszowego (i międzyznakowego jeśli to możliwe). Programy w których można zmieniać odstęp międzywierszowy skokowo rzadko dają się dopasować do stawianych im szczegółowych wymagań.







6. 75% możliwości Twojego edytora będzie zależnych od drukarki jaką kupisz lub jaką już dysponujesz. Uczulam Cię więc, że na drukarce w żadnym wypadku NIE WARTO OSZCZĘDZAĆ. Jeśli będzie to model okrojony z większych możliwości, to wydrukowany tekst będzie dokładnym odbiciem zaawansowania technicznego drukarki. W chwili obecnej są dostępne na rynku tanie i bardzo dobre drukarki o bardzo wielkich możliwościach, nie idź więc na łatwiznę kupując antyki z lat osiemdziesiątych.
7. W każdym edytorze powinny być opcje działania na blokach tekstu (np. kopiowanie, przenoszenie), możliwość definiowania nagłówków i stopek, regulowana tabulacja, wcięcia tekstu; w edytorach wyższej klasy spotyka się kolumnowy tryb pracy oraz możliwość tworzenia tabelek. W pracy niezbędne będą również takie funkcje jak wyszukiwanie tekstu, wyszukiwanie i zastępowanie, operacje dyskowe.
8. Aby zapobiec błędom ortograficznym edytor powinien mieć dołączony słownik zawierający możliwie największą liczbę słów; taka opcja, zwana zwykle SPELL CHECK, jest bardzo pomocna przy usuwaniu literówek. Dodam jeszcze, że porządny edytor pozwala użytkownikowi tworzyć swoje własne słowniki.
   Podobną rolę pełni tzw. THESAURUS zawierający wyrazy bliskoznaczne co pozwala na zróżnicowanie Twojego stylu pisania i wyeliminowanie często powtarzających się wyrazów i zwrotów.
9. Jeśli korzystasz z edytora do prac czysto biurowych, to pamiętaj, że byłoby dobrze, aby współpracował on z jakimś arkuszem kalkulacyjnym i bazą danych. W ten sposób możesz tworzyć okólniki, wystawiać np. faktury bez kłopotliwego przenoszenia danych itp. Często pożądane jest, aby Twój edytor pozwalał na łączenie grafiki z tekstem.
10. Dobry edytor poznasz również po dużej liczbie translatorów dokonujących konwersji dokumentów na formaty innych edytorów. Bardzo ważna jest tu opcja SEQUENTIAL FILE (plik sekwencyjny), gdyż tekst w tej postaci potrafią wczytywać i przetwarzać prawie wszystkie edytory dla ośmiobitowych Commodore. Z polskimi literami (jeśli je masz) można sobie dość łatwo poradzić — zastąp je przed zapisem za pomocą dwóch znaków (np. wykrzyknik i a zastępują literę „ą").

Czas na krótką charakterystykę każdego z edytorów. W swoich rozważaniach pozwolę sobie na pominięcie edytora Polscript, który omówił za mnie na łamach tego numeru „C&A" Bartek Dramczyk.

FONTMASTER II

Commodore 64

Zdecydowanie najlepszy i najbardziej uniwersalny edytor jaki spotkałem pracujący w trybie GRAFICZNYM dla Commodore 64. Jest on wyposażony w kilkadziesiąt świetnie wykonanych krojów pisma (można nim pisać m.in. po hebrajsku), edytor czcionek, wiele unikalnych opcji dotyczących formatowania tekstu (np. możesz utworzyć maksymalnie cztery niezależne kolumny tekstu). W trakcie pisania możesz wykorzystać jednocześnie do 9 różnych czcionek. FONTMASTER jest wyposażony w opcję PREVIEW uwzględniającą UKŁAD tekstu na stronie i podział dokumentu na strony. Nie pozwala na łączenie grafiki z tekstem. Edytor ten nadaje się znakomicie do tworzenia wydruków o bardzo wysokiej jakości, estetyce i elegancji. Wersja dla C-64 nie ma możliwości korzystania ze słownika (SPELL-CHECK), ma ją za to wersja dla C-128. Ta ostatnia różni się od programu dla C-64 tylko większą liczbą krojów pisma oraz ogólnymi parametrami wynikającymi głównie z różnic pomiędzy C-64 i C-128. Do absolutnie spartaczonych należy opcja PREVIEW — jest to najgorsza wada wersji dla C-128.

WYMAGANIA:

Drukarka wysokiej klasy mająca możliwość pracy w trybie graficznym z poczwórną gęstością (quadruple density), stacja dysków. Wydruk przebiega bardzo wolno, nie jest to więc edytor dla niecierpliwych. W wersji dla C-128 pracuje w trybie 80 znaków na ekranie.

GEOWRITE

Commodore 64 i Commodore 128

Także edytory pracujące w trybie GRAFICZNYM, mające duże możliwości związane projektowaniem własnych znaków i stosowaniem wielu krojów pisma. W przeciwieństwie do Fontmastera II wpisywanie tekstu przebiega całkowicie w trybie graficznym, jest to więc edytor wolniejszy i chyba nieco bardziej prymitywny. Jego główną zaletą jest duża liczba czcionek, które można dowolnie mieszać na ekranie oraz możliwość łączenia tekstu z grafiką (pliki programu GEOPAINT). Jest to edytor WYSIWYG (What You See Is What You Get — to co widzisz na ekranie jest tym co otrzymasz na papierze).

GEOWRITE współpracuje z ogólnie znanymi programami działającymi pod kontrolą otoczenia GEOS (np. GEOMERGE, GEODEX, GEOFILE), w tym także słownikiem znanym pod nazwą GEOSPELL. Ogólnie rzecz biorąc jest to edytor wolniejszy, o możliwościach zbliżonych do Fontmastera, jednakże pod wieloma względami jest on bardziej prymitywny. Wady te są rekompensowane współpracą edytora z oprogramowaniem działającym w otoczeniu GEOS.

WYMAGANIA:

Jeżeli chcesz uzyskać wydruki wysokiej klasy musisz mieć drukarkę o parametrach jak dla edytora Fontmaster. Ponieważ GEOS nie pracuje z magnetofonem wymagana jest również stacja dysków.

PRINTFOX

Commodore 64

Kolejny edytor pracujący w trybie graficznym. Do jego głównych zalet należy przede wszystkim możliwość łączenia grafiki z tekstem (biblioteki rysunków i rysuneczków liczą sobie obecnie kilka lub nawet kilkanaście tysięcy obrazków).

Stosowany głównie w celach związanych z DTP (DeskTop Publishing) czyli składaniem gazetek, ogłoszeń, zaproszeń itp. Edytor ten zostanie niebawem omówiony dokładniej na łamach „C&A".

WYMAGANIA:

Jak każdy edytor graficzny PRINTFOX jest dość powolny i wymaga drukarki co najmniej średniej klasy. Może być wykorzystany do sporządzania wydruków o wysokiej jakości, pod warunkiem zapewnienia mu odpowiedniej drukarki.

TEXTOMAT (TEXTOMAT PLUS)

Commodore 64 i Commodore 128

Jeden z edytorów, które na ekranie pracują w trybie znakowym natomiast na drukarce — w trybie graficznym. Jego popularność wynika przede wszystkim z bardzo prostego i użytecznego przypisania polskich znaków. W działaniu jest dość powolny (zwłaszcza opcja wstawiania bloku tekstu), nawet na C-128. Zawiera kilka różnych zestawów znaków ekranowych oraz dla drukarki. Nie spotkałem się z edytorem czcionek w wersji dla Commodore 64, jest on natomiast dostępny w wersji dla Commodore 128. Do ciekawszych opcji (wersja 128) należy możliwość transmisji tekstu przez modem akustyczny, łączenie grafiki z tekstem w ograniczonym zakresie. Wydruk nie ma płynnej regulacji odstępu międzywierszowego (przynajmniej dla drukarek Star serii LC-10). Wiem również, że swego czasu edytor ten był przystosowywany do wydruku polskich liter w trybie korespondencyjnym (NLQ) nie miałem jednak dostępu do tej wersji programu. O ile mi wiadomo nie współpracuje z żadnymi słownikami.

WYMAGANIA:

Drukarka z możliwością pracy w trybie graficznym, stacja dysków.

FLEET SYSTEM III

Commodore 128

Jeden z najlepszych edytorów tekstu jaki znam dla Commodore 128 — pod warunkiem, że oceniamy go w grupie programów bez możliwości korzystania z polskich liter. Przez długi okres czasu był on przeze mnie wykorzystywany do wpisywania i przetwarzania tekstów angielskich. Szybki w pracy i wydruku (pracuje tylko w trybie 80-znakowym C-128), współpracuje z bardzo dużym objętościowo słownikiem ortograficznym i THESAURUSEM. Współpracuje również z bazą danych FLEET FILER tworząc z nią interesujący zestaw programów.

WYMAGANIA:

Commodore 128 pracujący w trybie 80-znakowym, stacja dysków.

EDYTOR PL

Commodore 64

Jest to edytor pracujący w oparciu o znaki przenoszone do pamięci RAM drukarki (download). Może współpracować tak ze stacją dysków jak i z magnetofonem (tzn. może zapisywać pliki na taśmie). Program ten współdziała również ze słownikiem ortograficznym. Po sporządzeniu własnych zestawów znaków (wymagana jest tu pewna wiedza) może służyć do sporządzania wydruków o wysokiej jakości. Edytor ten jest dopasowany do bardzo popularnych w Polsce drukarek Star serii LC-10 i kompatybilnych.

WYMAGANIA:

Commodore 64, stacja dysków, jeśli chcesz mieć wydruki wysokiej klasy — dobra drukarka. Program współpracuje również z tańszymi drukarkami, np. MPS-803 na której również można uzyskać polskie znaki.

Podane tu przeze mnie typy nie wyczerpują w żadnym wypadku tematu; liczę, że i nasi Czytelnicy coś dorzucą na ten temat, ponieważ wykorzystanie komputera do przetwarzania tekstu jest jednym z podstawowych celów stawianym tym maszynom. Zapraszam zatem na łamy, a za miesiąc spotkamy się tym razem z opisem edytora FONTMASTER.

KLAUDIUSZ DYBOWSKI

MON 64



Dzięki uprzejmości naszego Czytelnika, pana Mariusza Siwca otrzymaliśmy do oceny moduł o nazwie MON 64. Jego historia sięga początków Commodore 64 — w latach 1983–1984 była to najlepsza karta z rozbudowanym monitorem języka maszynowego.

Sam program monitora zajmuje 8 KB pamięci i może być umieszczony w RAM albo od adresu $8000, albo też od $A000 (wyboru dokonuje się przed włączeniem komputera za pomocą przycisku na obudowie karty). Oczywiście, jeśli wybierzemy obszar od $A000, nie mamy możliwości korzystania z interpretera BASIC — jest on definitywnie wyłączony. Problem ten nie występuje, gdy monitor zajmuje obszar od $8000. Wtedy, po włączeniu komputera ukazuje się normalny obraz z komunikatem o wolnej pamięci, napisem READY i kursorem.

Przejście do monitora zapewnia kombinacja klawiszy RUN/STOP i RESTORE, a powrót do BASIC — rozkaz .X. Nie ma więc problemu z układaniem i natychmiastowym sprawdzaniem działania programów mieszanych, wykorzystujących zarówno BASIC, jak i język wewnętrzny.

Istotną zaletą modułu MON 64 jest opcja umożliwiająca wybór pomiędzy RAM i ROM (.*). Dzięki niej użytkownik ma do dyspozycji praktycznie pełne 64 KB wolnej pamięci na swoje potrzeby. MON 64 wykorzystuje po prostu „naturalną" cechę C-64, jaką jest bankowanie pamięci (RAM i ROM znajdują się w tym samym obszarze, ale nie można z nich korzystać jednocześnie). Omawiana opcja, w połączeniu z rozkazami SAVE (.S) i LOAD (.L) umożliwia np. zapisanie na nośniku zawartości ROM, a następnie jej przywołanie do pamięci operacyjnej i dowolną modyfikację. Postępując w ten sposób, tj. wykorzystując gotowe procedury z ROM i ograniczając się jedynie do wprowadzania poprawek, można bardzo łatwo tworzyć sobie własne, nowe procedury obsługujące przerwania, nowy interpreter BASIC, czy też — jeśli ktoś się uprze — nowy system operacyjny.

Ciekawą opcją jest WALK EXECUTION (.W), która pozwala śledzić krok po kroku działanie gotowego już programu czy procedury. Rozkazy wyświetlane są na ekranie w kolejności wykonywania, a więc np. po odgałęzieniach typu BNE, BEQ itp. lub po skokach JMP (JSR) programista fizycznie widzi, co komputer wykonuje. Oczywiście podczas wyświetlania rozkazów program jest normalnie wykonywany. Przejście do następnego rozkazu musi zaakceptować użytkownik wciśnięciem klawisza „P". Aby wyjść z trybu WALK EXECUTION wystarczy nacisnąć klawisz STOP.

Często chcielibyśmy zobaczyć, jak zmienia się zawartość danej komórki pamięci w trakcie działania programu. Służy do tego rozkaz .V. Adres, jaki wpiszemy po tej instrukcji, ukaże się w prawym górnym rogu ekranu, a jego zmieniająca się zawartość — pod spodem. Rozkaz ten działa tylko w trybie WALK EXECUTION.

Relokowanie (przemieszczanie) programów, nawet tych, które posiadają rozkazy skoków bezwzględnych (JMP, JSR), nie sprawia żadnych kłopotów. Najpierw należy użyć rozkazu TRANSFER (.T), który przenosi program mechanicznie z jednego miejsca w drugie, a potem rozkazem .N dokonujemy odpowiedniej zmiany adresów przypisanych JMP i JSR.

Inne ułatwienia, jakie oferuje użytkownikowi MON 64, to możliwość bezpośredniego ustawiania rejestrów procesora (F3), tryb QUICK-TRACE (wykonywanie programu w zwolnionym tempie; podobne do WALK EXECUTION, ale bez wyświetlania rozkazów) oraz możliwość wpisywania tekstu bezpośrednio do pamięci (INTERPRET).

Dla początkujących przewidziano dwie instrukcje (.#, .$) powodujące konwersję liczb z DEC na HEX i odwrotnie. Ponadto można włączać/wyłączać powtarzalność klawiszy (F5, F6), jest też ekran pomocniczy z krótkim opisem wszystkich funkcji.

Jeśli chodzi o współpracę modułu z drukarką, to dostępne są dwa polecenia: DUMP SCREEN (F1), czyli kopia aktualnego ekranu oraz OUTPUT TO PRINTER (.O#4) — po wybraniu tej opcji wszystko, co normalnie byłoby wyświetlane na ekranie, zostanie wydrukowane.

Nieco gorzej przedstawia się współpraca MON 64 ze stacją dysków. Działanie tego urządzenia nie jest ani o włos szybsze niż normalnie. MON 64 nie dysponuje też pełnym zestawem poleceń (brak RENAME, SAVE&REPLACE i VERIFY). Ale to właściwie jedyna poważniejsza wada modułu wynikająca raczej z faktu, że program powstawał w czasach w których za jedną stację można było kupić dwa komputery.

CGA

OD REDAKCJI:

Artykuł ten nie będący w rzeczywistości ani raportem ani testem, lecz materiałem poglądowym powstał dzięki uprzejmości naszego Czytelnika, który zdecydował się wypożyczyć nam MON 64 na kilka tygodni. Panu Mariuszowi Siwcowi serdecznie dziękuję, a innych Czytelników zapraszam do kontynuowania takiej współpracy (po zapoznaniu się z urządzeniem natychmiast je odsyłamy). Zainteresowanych nabyciem MON-64 informuję, że być może jest to jeszcze możliwe w Szwecji. Adres dystrybutora podam wszystkim zainteresowanym po nadesłaniu do redakcji koperty zwrotnej ze znaczkiem.



POLSCRIPT

Jednym z wielu zastosowań komputerów jest edycja i przetwarzanie tekstów. Użytkownicy komputerów IBM mają do dyspozycji kilkadziesiąt edytorów, posiadacze Amig mogą korzystać z Cygnusa i Polscribbie'a. A C-64? Czy ten komputer nie potrafi przetwarzać tekstu? Ależ skąd — istnieje wiele znakomitych edytorów dla C-64. Jednym z nich jest POLSCRIPT. I o nim właśnie będzie ten artykuł.

Redagowanie czasopisma takiego jak np. „C&A" łączy się z przetwarzaniem niesamowitych ilości tekstu. Jednego dnia 30 kB, drugiego 100 kB... Przy takiej pracy program, za pomocą którego tekst jest przetwarzany, powinien charakteryzować się wygodą obsługi. Chodzi o to, żeby człowiek po spędzeniu 10 godzin nad edytorem w dalszym ciągu wiedział nie tylko jak się nazywa, lecz również pamiętał, że np. ALT+Q to wyjście z programu, a nie przesunięcie kursora do końca strony. Tej cechy nie brakuje w POLSCRIPCIE.

Pierwowzorem tego edytora był równie stary jak C-64 program EASY SCRIPT. Potem ktoś wpadł na pomysł, żeby dorobić do niego polskie znaki i w efekcie powstał jeden z lepszych edytorów na C-64. POLSCRIPT ma wiele funkcji ułatwiających życie użytkownikowi. Wydruki wyglądają całkiem przyzwoicie nawet po wykonaniu ich na stosunkowo prostych i tanich drukarkach. Niestety drukowanie nawet jednej strony to katorga — POLSCRIPT wykorzystuje tryb graficzny drukarki, co zajmuje dużo czasu. Edytor ten nadaje się bardzo dobrze tak dla właścicieli stacji dysków jak i posiadaczy MAGNETOFONÓW — może on być wczytywany z taśmy czy dyskietki, może również zapisywać dokumenty na obu nośnikach. Tworzone pliki tekstowe są typu sekwencyjnego, co umożliwia ich swobodne przenoszenie za pomocą programu BIG BLUE READER na format ASCII IBM. W swojej pracy korzystam również z nieco bardziej zaawansowanych programów (stawiających jednocześnie wyższe wymagania sprzętowe) lecz POLSCRIPT darzę specjalnym sentymentem.

Po uruchomieniu programu ukazuje się plansza, na której należy zdefiniować podstawowe parametry programu. Najpierw należy ustalić liczbę znaków w wierszu (40–240). Nawet przy większej ilości znaków w wierszu niż 40, praca z tym edytorem jest całkiem wygodna. Drugie pytanie dotyczy pamięci zewnętrznej. Może to być stacja dysków (D) lub magnetofon (T). Teraz kolej na opis drukarki — POLSCRIPT poza drukarkami Commodore obsługuje również inne typy drukarek, nieprzystosowane firmowo do bezpośredniej współpracy z szyną Commodore 64. W takich wypadkach wymagany jest oczywiście odpowiedni interfejs.

Następnie program zadaje jeszcze dwa pytania: o zestaw znaków — angielski (E) lub polski (P) oraz o typ klawiatury — polską, lub angielską (polska klawiatura przeznaczona jest dla tych, którzy właśnie przesiadają się z maszyny do pisania).

Układ polskich liter nie jest taki, jak w innych edytorach tekstów (ALT lub CTRL i odpowiednia litera). Jest jednak dosyć łatwy w obsłudze i już po ok. pół godzinie można go opanować. A oto tabela znaków i odpowiadających im klawiszy:

ą —	;	Ł —	SHIFT+:
ć —	SHIFT+/	Ż —	SHIFT++
ę —	SHIFT+]	@ —	\
ł —	:	zł —	^
ń —	SHIFT+.	? —	SHIFT+6
ó —	@	+ —	SHIFT+−
ś —	SHIFT+,	: —	*
ź —	SHIFT+@	; —	SHIFT+*
ż —	−+	przełączanie znaków (ang-pol) —	SHIFT+^

Nie przedstawię wszystkich funkcji tego edytora z bardzo prostej przyczyny: nawet pobieżne omówienie wszystkich opcji zajęłoby pół numeru. W zamian postaram się w dużym skrócie omówić sposób korzystania z tego programu.

Górna linia programu stanowi okno informacyjne. Tutaj pojawiają się informacje o włączonych trybach, pozycja kursora, komunikaty programu. Autorzy przeróbki nie pomyśleli niestety o zamianie angielskich komunikatów systemowych na język polski i stąd są one wyświetlane w języku angielskim.

Kolor tekstu, tła i ramki zmienia się odpowiednio klawiszami: CTRL+1, CTRL+2, CTRL+3. Większość funkcji wywołuje się klawiszem F1+litera. Po ekranie można poruszać się tak, jak podczas pracy w trybie ekranowym. Po każdej wpisanej linii tekstu powinien znajdować się kod końca linii (RETURN) — w przeciwnym wypadku program może często „gubić" tekst. Klawisz F5 włącza i wyłącza duże litery.

Z polskich znaków są tylko dwie duże litery: „Ł" i „Ż". Klawisz F3 („*" w rewersie) określa polecenia związane z formatowaniem wydruku. Wiersze poprzedzone tym symbolem powinny znajdować się na początku tekstu. Klawisz F7 przesuwa kursor do następnego tabulatora poziomego (o ile taki został ustawiony), F8 — skok do następnego tabulatora pionowego. Klawisz F6 włącza tryb dziesiętny, przydatny np. podczas wprowadzania obliczeń do tekstu. Tryb ten można zaktywizować dla wybranej pozycji tabulatora. Przesuwanie znaków w lewo usuwa wszystkie spacje w wierszu na lewo od kursora.

Klawisz F4 umożliwia przejście do operacji dyskowych. Dostępne są następujące operacje:

$x — katalog dyskietki w stacji 0 lub 1,
Sx:nazwa — kasowanie zbioru (x- numer stacji, 0 lub 1)
Nx: nazwa, id — formatowanie dysku,
Ix — inicjalizacja stacji.

Uwaga!!! Nigdy nie zapisuj pliku tekstowego w formie @: nazwa — tekst w takim pliku ZAWSZE będzie uszkodzony.

F1+RUN/STOP — inicjalizacja (RESET) programu (z pytaniem o potwierdzenie) — radzę używać bardzo ostrożnie. Klawisze RUN/STOP-RESTORE umożliwiają powrót do ekranu konfiguracyjnego. Nie warto odtwarzać plików o innej ilości znaków w wierszu przy zmianie tej wartości. SHIFT+CLR/HOME umożliwia powrót do początku tekstu. F1+I włącza i wyłącza tryb wstawiania znaków.

F1+kursor w górę — ciągły ruch ekranu w górę.
F1+kursor w dół — ciągły ruch ekranu w dół.
F1+kursor w prawo — ciągły ruch ekranu w prawo.
F1+kursor w lewo — ciągły ruch ekranu w lewo. (RUN/STOP przerywa te operacje). Szybsze przesuwanie ekranu — SHIFT.
Pauza w przesuwaniu — spacja.

F1+INST/DEL — usunięcie linii tekstu.

F1+SHIFT+INST/DEL — wstawienie pustej linii.

F1+R/RETURN — zaznaczenie fragmentu od aktualnej pozycji kursora aż do naciśnięcia RETURN.

F1+D/RETURN — usunięcie fragmentu od aktualnej pozycji kursora aż do naciśnięcia RETURN.

F1+E+S — usunięcie zdania (od kropki do kropki).

F1+E+A — skasowanie całego tekstu.

F1+X — przeniesienie zaznaczonego fragmentu tekstu.

F1+A — powtórzenie zaznaczonego fragmentu tekstu.

F1+SHIFT+F+„nazwa" — zapisanie w pamięci zewnętrznej zaznaczonego fragmentu.

F1+L+„nazwa" — ładowanie zbioru tekstowego z pamięci zewnętrznej.

F1+I+F1+L+„nazwa" — dołączenie zbioru z pamięci zewnętrznej do tekstu poddawanego edycji (od miejsca w którym jest aktualnie zlokalizowany kursor).

F1+F+„nazwa" — zapis tekstu poddawanego edycji.

strzałka w lewo — przesunięcie do końca poprzedniej linii.

F1+spacja — następna strona.

F1+SHIFT+spacja — poprzednia strona.

F1+G+E — przesunięcie na koniec tekstu.

F1+G+xxx — skok do linii xxx.

F1+T+H — ustawienie tabulatora poziomego (pozycja kursora).

F1+T+V — ustawienie tabulatora pionowego (pozycja kursora).

F1+C+H — kasowanie tabulatora poziomego (pozycja kursora).

F1+C+V — kasowanie tabulatora pionowego (pozycja kursora).

F1+Z+H — kasowanie wszystkich tabulatorów poziomych.

F1+Z+V — kasowanie wszystkich tabulatorów pionowych.

F1+P — wskazanie tabulacji poziomej.

F1+S+„stary"+„nowy" — deklarowanie znaku (znaków) do wyszukania i/lub zastąpienia.

F1+H+M — wyszukiwanie zadanego ciągu znaków w pamięci.

F1+H+L — wyszukiwanie zadanego ciągu znaków w pliku.

F1+@+M — wyszukiwanie i zastępowanie zadanego ciągu znaków w pamięci.

F1+@+F — wyszukiwanie i zastępowanie zadanego ciągu znaków w pliku.

Polecenie sterujące formatem wydruku (po symbolu F3) powinny być oddzielane od siebie symbolem „:". Linię musi kończyć znak „;".

JU1 — justyfikacja włączona.
JUO — justyfikacja wyłączona.
RAx — wyrównanie do prawej (1–wł, 0–wy).
CNx — centrowanie tekstu (1–wł, 0–wy).
LMxx — ustawienie lewego marginesu.
RMxx — ustalenie prawego marginesu.
OFxx — przesunięcie w prawu obu marginesów o xx odstępów.
FPO — następna strona.

Myślę, że ta ściągawka pomoże posiadaczom tego programu w jego lepszym wykorzystaniu. Opisane powyżej funkcje i polecenia w zupełności wystarczają do pracy z tym edytorem. Trzeba przyznać, że Polak jak chce, to potrafi. LET'S POLSCRIPT.

BAD

PROTECTOR V1.0

UWAGA: Ze względu na objętość program opisywany w tym artykule jest rozpowszechniany WYŁĄCZNIE na dyskietce „C&A" nr 07/92.

W trzecim numerze „Commodore & Amiga" prezentowany był program mojego autorstwa — DISK CODER. Służył on do zabezpieczania dysków w taki sposób, aby niepowołane osoby nie mogły nic z niego odczytać.

Program ten nie był jednak pozbawiony wad; na przykład po odkodowaniu dysku trzeba było go ponownie zakodować. Czas trwania jednej z tych operacji wynosił w przybliżeniu około 28 minut. Te wady skłoniły mnie do napisania zupełnie nowego programu, służącego do zabezpieczania Twoich dyskietek. Nowa wersja działa na zupełnie nowej zasadzie. Koduje ona wszystkie pliki na dyskietce z adresem początkowym 2049 ($0801), czyli te programy, które wczytujemy bezpośrednio do komputera i uruchamiamy rozkazem RUN.

Po zakodowaniu program zapisuje na zabezpieczaną dyskietkę plik, pozwalający na zdekodowanie, czyli na uruchomienie danego programu (tylko wtedy, gdy podasz właściwe hasło). Procedura wczytująca i dekodująca jest wyposażona w turbo przyspieszające wczytywanie programu do zdekodowania.
W programie dekodującym wykorzystałem turbo z MEANING OF LIFE grup THE 2ND SOCIETY i $4753.
Obsługa programu dekodującego jest banalnie prosta. Po jego uruchomieniu komputer pyta się o hasło. Po jego wpisaniu komputer wyświetla katalog dyskietki (FIRE w joysticku podłączonym do portu 2 zatrzymuje wyświetlanie katalogu). Teraz wskaż strzałką nazwę programu do zdekodowania i naciśnij przycisk FIRE. Komputer automatycznie wczyta program z dyskietki (w turbo), zdekoduje go i uruchomi. Jeśli chcesz jeszcze raz wyświetlić katalog dyskietki (podczas wyboru programu do dekodowania), wskaż strzałką ikonę z napisem „DIR" i naciśnij przycisk FIRE. Po naciśnięciu klawisza RESTORE program powróci do procedury wpisywania hasła.
Aby zakodować dyskietkę, uruchom program i wpisz wybrane hasło. Po jego wpisaniu na ekranie ukaże się menu, z którego można wybrać opcje wyświetlenia katalogu dyskietki lub przejścia do kodowania plików (odpowiednio ikony „DIR" i „START"). Naciśnięcie klawisza RESTORE powoduje ponowne uruchomienie programu, tj. powrót do wpisywania hasła. Po wybraniu ikony „START" komputer będzie kolejno wczytywał wszystkie pliki typu PRG o adresie wczytywania 2049 ($0801), a następnie kodował je. Po zakodowaniu zbiór znajdujący się aktualnie w pamięci zostanie skasowany, a jego zdekodowana wersja na dyskietce będzie zastąpiona wersją zaszyfrowaną. Operacja kończy się zapisem na dyskietce programu dekodującego. Opisany tu program świetnie nadaje się do kodowania wszelkich dyskietek — nawet tych, na których zakładane są pliki typu SEQ (PROTECTOR nie koduje zbiorów typu SEQ, REL, USR, DEL oraz PRG o innym niż 2049 ($0801) adresie.
Na zakończenie kilka spraw, o których warto pamiętać podczas korzystania z niniejszego programu:

Dyskietka nie może być zabezpieczona przed zapisem ani w sposób sprzętowy, ani w sposób programowy. Pamiętaj również, że struktura sektora 0 na ścieżce 18 nie może być w żaden sposób zmieniona.

Na dysku zabezpieczanym musi pozostać minimum 15 bloków wolnych (na program dekodujący).

Pliki do zakodowania nie mogą być zabezpieczone przed skasowaniem (znak „<" w katalogu dyskietki).

Jeżeli będziesz kodował jakieś ważne dla Ciebie programy, zrób sobie NIE ZAKODOWANĄ kopię bezpieczeństwa.

Jeśli chcesz, aby jakieś pliki typu PRG o adresie początkowym 2049 ($0801) nie zostały zakodowane zabezpiecz je przed skasowaniem (widoczny znak „<" przy typie takiego pliku).

Podczas wpisywania hasła w obecności osób postronnych można przed tą operacją zmienić kolor tekstu na kolor zgodny z tłem ekranu (CTRL-1). W ten sposób wpisywane hasło nie będzie widoczne.

MARIUSZ FERDYN

ASEMBLER 6502 – część II

W tym odcinku zajmiemy się już wyświetlaniem całych wyrazów, a nawet zdań. Najpierw jednak postaraj się, by w komórce $2000 (czyli 8192) znalazła się wartość 1. Jak — to już Twój problem. Sugerowałbym jednak małe POKE...

POKE 8192, 1

I po co nam to było? To proste — dzięki temu mamy w komórce pamięci o adresie 8192 zapisaną wartość i dzięki czemu możemy ją pobrać w dowolnym momencie. Ale jak? — zapyta spostrzegawczy Czytelnik. Posłużymy się rozkazem LDA. Tym razem jednak wartość po rozkazie nie będzie oznaczać LICZBY, którą chcemy wpisać do akumulatora, lecz ADRES komórki, w której będziemy szukać LICZBY, wpisanej do akumulatora. Wpisz więc programik przedstawiony poniżej:

Monitor	Asembler
	* = $2710
A 2710 LDA $2000	LDA $2000
A 2713 STA $0400	STA $0400
A 2716 RTS	RTS
G 2710	strzałka w lewo, 3, s

No i proszę, efekt zgodny z tym, co przed chwilą napisałem. Jeżeli przed liczbą po rozkazie LDA nie wpiszemy znaczka #, to komputer będzie traktował tę liczbę jako adres komórki pamięci w której jest zapisana liczba przeznaczona do wczytania do akumulatora.

Teraz zajmijmy się obiecanymi przedtem słowami. Jak zapewne zdążyłeś zauważyć, wpisywanie słów dłuższych, niż dwu-trzyliterowe metodą LDA – STA przestaje być rozrywką i staje się średnio przyjemnym wklepywaniem kolejnych liczb. Jeśli umiesz programować w jakimś innym języku, zapewne przypominasz sobie jakieś pętle: FOR...NEXT, REPEAT...UNTIL, DO...LOOP i inne. Jeśli zaś niczego sobie nie przypominasz, to wiedz, że pętla, to taki chytry chwyt, dzięki któremu możesz napisać jakiś ciąg instrukcji raz, a potem powtarzać go kilka razy. Odbywa się to nie za pomocą przepisywania tego samego fragmentu programu na nowo, lecz poprzez poinformowanie komputera, że ma wykonywać tę część programu tyle razy, ile sobie zażyczysz.

W asemblerze jest kilka metod zrobienia pętli. Możemy na przykład wstawić do jednej z komórek liczbę powtórzeń i po każdym przebiegu pętli zmniejszać jej wartość o 1. Po dojściu do 0 — kończymy powtarzanie pętli. Znacznie wygodniejszym rozwiązaniem jest jednak wykorzystanie jednego z DODATKOWYCH REJESTRÓW procesora. Bowiem procesor ma nie tylko akumulator. Ma także rejestry oznaczone jako X i Y. Za pomocą np. rejestru X możemy także utworzyć pętlę i to znacznie lepszą niż przy pomocy komórki pamięci. Sama zasada będzie oczywiście taka sama. A oto przykład takiej pętli — wydrukujemy kilka liter za pomocą jednego STA!

Monitor	Asembler
	* = $2710
A 2710 LDA #$02	LDA #2
A 2712 LDX #$10	LDX #$10
A 2714 STA $0400, X	U STA $0400, X
A 2717 DEX	DEX
A 2718 BNE $2714	BNE U
A 271A RTS	RTS
G 2710	strzałka w lewo, 3, s

Wszystko się zgadza? Mamy aż szesnaście literek „B", a rozkazów wydaliśmy, jak dla trzech! Pojawiły się jednak nieznane Ci rozkazy. Zaraz je wyjaśnię.

LDX #xx — weź liczbę xx i wpisz ją do rejestru X (LoaD X register)
DEX — zmniejsz zawartość rejestru X o 1 (DEcrement X register)
STA xxxx,X — zapisz zawartość akumulatora do komórki o adresie xxxx, zwiększonym o zawartość rejestru X. (STore Accumulator)
BNE xxxx — jeżeli ostatnia wykonana operacja nie przyniosła wyniku zerowego, wykonaj skok do adresu xxxx. (Branch, if Not Equal to zero)

Posiadacze asemblera mają właśnie okazję, by dowiedzieć się, czym właściwie jest etykieta. Popatrz tylko, co wpisałeś po rozkazie BNE — tylko literkę U. A co powinieneś wpisać? Adres, do którego komputer ma wykonać skok. Wniosek — etykieta służy nam do określenia adresu bez potrzeby dokładnej jego znajomości. Takie samo U postawiliśmy przed rozkazem STA. Komputer porównuje jedną etykietę z drugą i, jeśli się zgadzają, wykonuje skok.

Przejdźmy do obiecanych całych słów. Przedtem jednak gorąco apeluję, by wszyscy zaopatrzyli się w TurboAssembler 5.1. Jak się zaraz przekonamy, ułatwi to wpisywanie programów znacznie bardziej, niż używanie jakiegokolwiek monitora. Inne makroasemblery mają inne, sobie tylko właściwe, możliwości i mogą z tego wyniknąć nieporozumienia.

Teraz jednak posiadacze monitorów muszą się posłużyć językiem BASIC. Chodzi tu o to, aby tekst do wyświetlenia na ekranie wpisać do określonego obszaru pamięci skąd zostanie on pobrany i wyświetlony na ekranie przez program napisany w języku maszynowym.

```
10 A$ = „PRAWDZIWY MANIAK PETLI SIE NIE BOI"
20 FOR A=1 TO LEN(A$)
30 B=ASC(MID$(A$, A, 1))
40 IF B>64 THEN B=B–64
50 POKE 8191+A, B
60 NEXT
```

Monitor	Asembler
	* = $2710
A 2710 LDX #$2C	LDX #$2C
A 2712 LDA $2000, X	WPAK LDA TEKST, X
A 2715 STA $0400, X	STA $0400, X
A 2718 DEX	DEX
A 2718 BNE $2712	BNE WPAK
A 271A RTS	RTS
	TEKST .TEXT „PRAWDZIWY MANIAK PETLI SIE NIE BOI"
G 2710	strzałka w lewo, 3, s

No i mamy obiecane całe wyrazy i zdania. I tylko jeden nowy rozkaz:
LDA xxxx, X — wstaw do akumulatora zawartość komórki o adresie xxxx, zwiększonym o zawartość rejestru X.

Użytkowników asemblera informuję, że instrukcja .TEXT pozwala na wpisanie w cudzysłowie jakiegokolwiek tekstu. W chwilach wolnych proponuję zastanowić się nad innym zagadnieniem: jak wyświetlić ten tekst od końca?

BARTŁOMIEJ I. KACHNIARZ

UWAGA:
Do wpisywania wszystkich zamieszczonych w tym artykule NIEZBĘDNY jest program typu MONITOR lub ASEMBLER."

DZIŚ POZNALIŚMY INSTRUKCJE:

LDA xxxx — wstaw zawartość komórki o adresie xxxx do akumulatora
LDA xxxx, X — wstaw zawartość komórki o adresie xxxx, zwiększonym o zawartość rejestru X do akumulatora
STA xxxx, X — zapisz akumulator do komórki o adresie xxxx, zwiększonym o zawartość rejestru X
LDX #xx — wstaw liczbę xx do rejestru xx
DEX — zmniejsz zawartość rejestru X o 1
BNE xxxx — skocz do komórki o adresie xxxx, jeżeli wynik ostatniej wykonanej operacji nie był równy 0

ZASILACZ do COMMODORE 64

Zasilacze fabryczne do C-64 ulegają częstym awariom. Przyczyną tego stanu rzeczy jest zaprojektowanie zasilacza na tzw. „styk", tzn. producent nie pozostawił niezbędnego zapasu mocy. Efektem tego jest nadmierne nagrzewanie się zasilacza, potęgowane jeszcze przez kiepską wentylację (szczelnie zamknięta obudowa, elementy zalane żywicą). Pół biedy jeżeli wykorzystujemy sam komputer z magnetofonem, lecz co zrobić, jeśli przyłączyliśmy jeszcze inne przystawki? W tym przypadku fabryczny zasilacz nie wystarczy, musimy wykonać konstrukcję, opisaną w tym artykule.

Omówiony tu zasilacz jest zaprojektowany z rezerwą, tzn. jego moc jest ok. 25% większa, niż zasilacza fabrycznego. Jego wykonanie nie powinno nastręczyć trudności nawet początkującemu majsterkowiczowi. Koszt urządzenia mieści się w granicach 100–120 tysięcy złotych (wiele zależy od miejsca gdzie będziesz kupował podzespoły).

Układ można zmontować na płytce drukowanej, której schemat przedstawiono obok. Stabilizator scalony musi być wyposażony w radiator, najlepiej żeberkowy. Jeżeli całość chciałbyś umieścić w metalowej obudowie, to stabilizator można do niej przymocować. Aby polepszyć odprowadzanie ciepła, należy posmarować miejsce styku stabilizatora z radiatorem (obudową) cienką warstwą pasty silikonowej.

Jeżeli zastosujemy radiator o zbyt małej powierzchni, to zadziała wewnętrzne zabezpieczenie termiczne układu stabilizatora ograniczające prąd wyjściowy. Może się to objawić w postaci błędnej pracy lub braku pracy komputera. W tym przypadku należy oczywiście zwiększyć powierzchnię radiatora chłodzącego.

Elementy takie jak gniazdo bezpiecznika i wyłącznik, należy zamontować na jednej ze ścianek obudowy. Kondensatory o wartości 100 nF muszą być ceramiczne (bezindukcyjne) i najlepiej przylutować je wprost do końcówek stabilizatora (choć przewidziano na nie miejsce na płytce).

ZASADA DZIAŁANIA

Napięcie zmienne pobierane z sieci 220 V/ /50 Hz jest doprowadzane poprzez wyłącznik dwubiegunowy i bezpiecznik do uzwojenia pierwotnego transformatora. Z uzwojeń wtórnych otrzymujemy napięcie 9V prądu zmiennego 50 Hz (oba uzwojenia są jednakowe). Napięcie z jednego z uzwojeń doprowadzamy wprost do komputera, a z drugiego poddajemy prostowaniu dwupołówkowemu w mostku Graetza, złożonym z czterech diod prostowniczych. Następnie prąd stały poddawany jest filtracji (kondensator 4700 μF) i podany na wejście stabilizatora. Wewnątrz układu znajduje się stabilizator napięcia, zabezpieczenie termiczne oraz przeciwprzeciążeniowe. Pierwsze z zabezpieczeń zadziała, gdy temperatura struktury układu scalonego przekroczy 175 stopni, drugie — gdy prąd pobierany z wyjścia układu przekroczy 1.5 A. W tym miejscu należy się uwaga: zabezpieczenie przeciwprzeciążeniowe w momencie wystąpienia zwarcia (przeciążenia) nie odcina napięcia na wyjściu, lecz ustala prąd wyjściowy na poziomie maksymalnym tzn. 1.5 A! Zabezpieczenie termiczne natomiast obniża napięcie wyjściowe (maleje wtedy prąd pobierany) do poziomu takiego, przy którym nie następuje dalsze nagrzewanie się układu. Kondensatory ceramiczne dołączone do nóżek stabilizatora zapewniają poprawną pracę układu przy dużej odległości zasilacza do transformatora i komputera.

DOBÓR CZĘŚCI

Zamiast transformatora TS 20/1 można zastosować również inny, o oznaczeniu TS 20/3. Różnią się one między sobą tylko kształtem rdzenia (parametry mają identyczne). Diody BYP 401-50 mogą być zastąpione zachodnimi 1N4007, lub innymi podobnymi. Ważne, aby wytrzymywały napięcie 20 V i prąd 1 A. Zamiast kondensatora 4700 μF/25V można zastosować inny, o większym napięciu znamionowym lub większej pojemności (im większa pojemność tym lepszy współczynnik filtracji). Przy zakupie stabilizatora należy zwrócić uwagę na jego oznaczenie. Nie nadają się do naszych zastosowań stabilizatory o oznaczeniu 78L05 (maksymalny prąd wyjściowy 100 mA) oraz 78M05 (prąd wyjściowy 0.5 A). Należy zastosować układy o oznaczeniu 7805, np. MA 7805 (Tesla) lub UL 7505 (CEMI). Najczęściej jednak spotykane w sklepach są układy produkcji zachodniej o różnych literach przed cyframi, nam potrzebny jest układ bez liter pomiędzy cyframi. Wyjątkiem jest tu układ o oznaczeniu 78H05 o maksymalnym prądzie wyjściowym 5 A, jest on jednak rzadko spotykany. Cały układ najwygodniej przyłączyć do komputera i sieci za pośrednictwem przewodów, odciętych z uszkodzonego zasilacza.

URUCHOMIENIE UKŁADU

UWAGA:
W URZĄDZENIU WYSTĘPUJĄ NAPIĘCIA NIEBEZPIECZNE DLA ŻYCIA. W ZWIĄZKU Z TYM MONTAŻ I URUCHOMIENIE URZĄDZENIA POWINNY ODBYWAĆ SIĘ BEZWZGLĘDNIE POD OKIEM FACHOWCA!

Po zmontowaniu układu należy bezwzględnie sprawdzić poprawność połączeń przed przyłączeniem do sieci i sprawdzić miernikiem napięcia wyjściowe. Jeżeli nie stwierdziliśmy żadnych usterek w montażu, należy miernikiem uniwersalnym zmierzyć napięcia na wyjściu. Napięcie stałe 5V powinno zawierać się w przedziale 4.9–5.1 V, napięcie zmienne 9 V może wynosić nawet do ok. 12 V (jego amplituda spadnie po dołączeniu obciążenia czyli komputera). Jeżeli uzyskaliśmy przy pomiarze wyniki takie jak podano wyżej, możemy przyłączyć zasilacz do komputera.

UWAGA:
JAKIEKOLWIEK BŁĘDY W MONTAŻU MOGĄ SPOWODOWAĆ BARDZO POWAŻNE USZKODZENIA KOMPUTERA!

TS 20/1

4 x BYP 401-50

WTAT
200 mA

220 V~

W

4700 µF / 25V

100 nF ceram.

7805

100 nF ceram.

+5V

GND

9V~

A K
BYP 401-50

7805
1 2 3

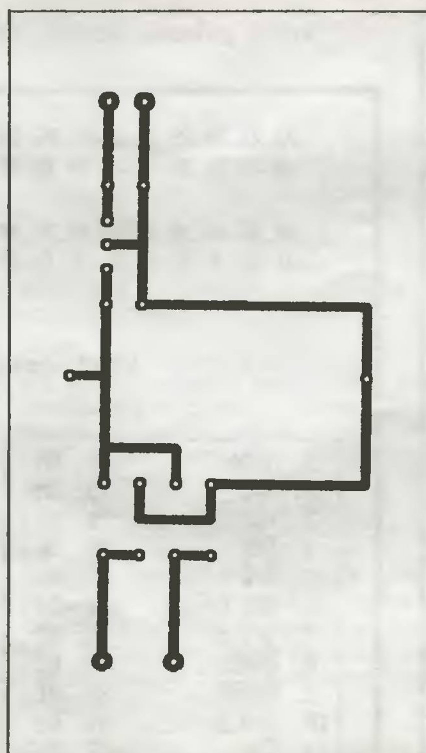
1-wejście
2-masa
3-wyjście

Schemat ideowy zasilacza do C-64 oraz rozmieszczenie końcówek elementów półprzewodnikowych.

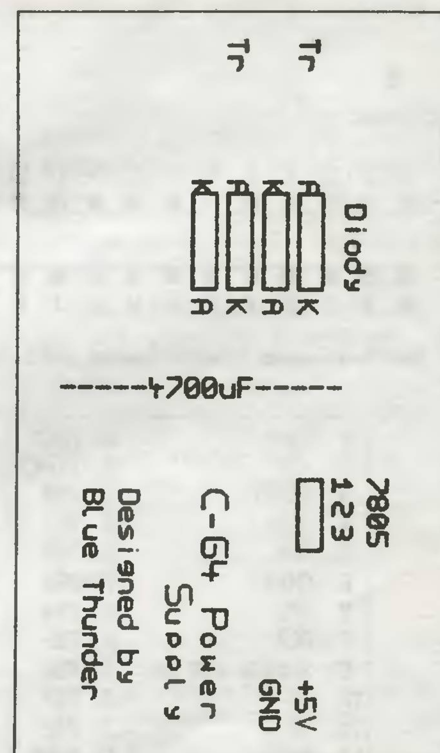
Wykaz elementów:

Transformator sieciowy TS 20/1 lub TS 20/3	1 szt.
Gniazdo bezpiecznika sieciowego wraz z wkładką topikową bezzwłoczną 200 mA	1 szt.
Wyłącznik dwubiegunowy np. typu Isostat	1 szt.
Dioda prostownicza BYP 401-50	4 szt.
Kondensator elektrolityczny 4700 µF/25 V	1 szt.
Kondensator ceramiczny 100 nF	2 szt.
Stabilizator scalony 5-woltowy 7805	1 szt.

JERZY „Blue Thunder" DUDEK



Strona lutowania (skala 2:1)



Rozmieszczenie elementów (widok od spodu płytki)

Przepraszamy!

W szóstym numerze "C&A" (strona 15) podaliśmy błędny numer telefonu do firmy TOMS. Stało się to powodem kłopotów dla wielu Czytelników oraz właścicieli podanego telefonu.

Poprawny numer to 27-16-01.

Jeszcze raz przepraszamy wszystkich Czytelników, firmę TOMS oraz posiadaczy podanego numeru telefonu.

Redakcja

RESET 64 na TRZY SPOSOBY

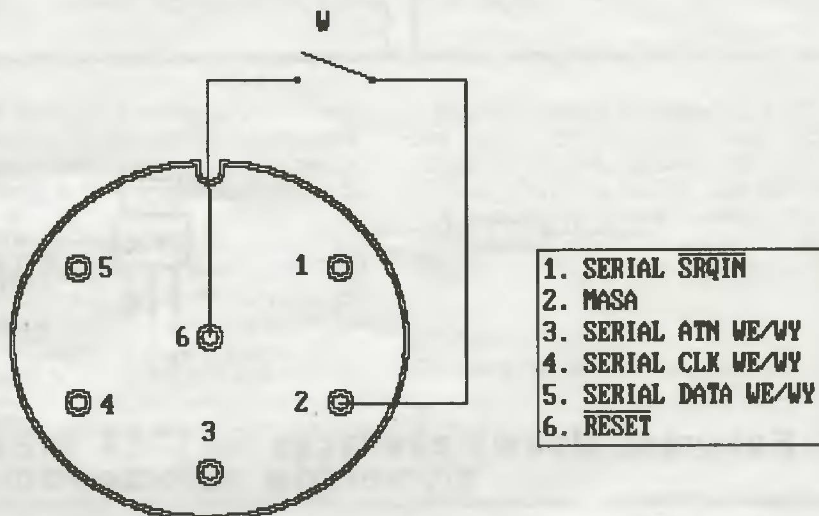
Commodore 64 ma pewien mały brak: nie zainstalowano w nim przycisku RESET. Niektóre ośmiobitowe komputery tej fimy np. C-16/116/128 itp. mają takie przyciski, lecz w przypadku C64 jest to po prostu niedopatrzenie.

Ten poważny brak nadrabiają oczywiście różnorakie moduły typu FINAL, ACTION REPLAY, X i inne. Co jednak zrobić, gdy w karcie siedzącej w MEMORY EXPANSION PORT (w porcie rozszerzenia) producent nie umieścił przycisku RESET, a autorzy książek poświęconych C-64 straszą, że częste wyłączanie i włączanie komputera efektywnie skraca jego „życie". Oczywiście istnieją rozwiązania zastępcze i to aż trzy.

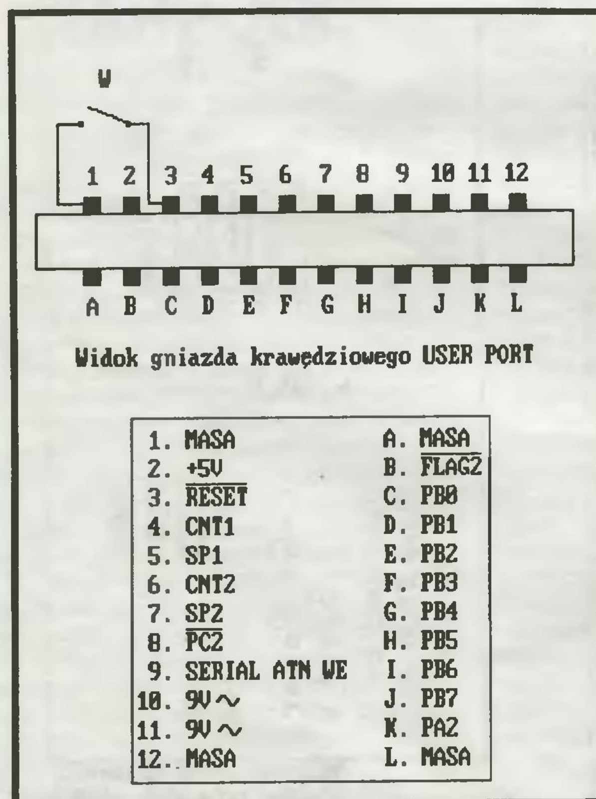
Pierwsze z nich (rysunek 1) to urządzenie przyłączone do portu szeregowego SERIAL. Rozwiązanie to było już wielokrotnie publikowane. Mikrowyłącznik astabilny łączy linie 6 i 2 tego portu. Wciśnięcie mikrowyłącznika spowoduje zwarcie sygnału RESET z masą (GND) i w efekcie inicjalizację komputera. Tu uwaga: nie w każdym Commodore 64 w porcie szeregowym występuje sygnał RESET!

W takiej sytuacji użytkownik może zastosować jedno z dwóch poniżej opisanych rozwiązań. Ponieważ linie RESET występują w porcie użytkownika oraz w porcie MEMORY EXPANSION jest rzeczą wyboru które z nich wybierzesz. Zwracam jednak uwagę, że znacznie bezpieczniejsze jest dołączenie wyłącznika do portu użytkownika ze względu na większe odstępy pomiędzy ścieżkami. Połączenia we wszystkich wypadkach należy wykonać bardzo starannie i uważnie; jeżeli nie czujesz się na siłach aby wykonać jedno z tych połączeń, zwróć się do kogoś kto to potrafi.

PIOTR LISZEWSKI

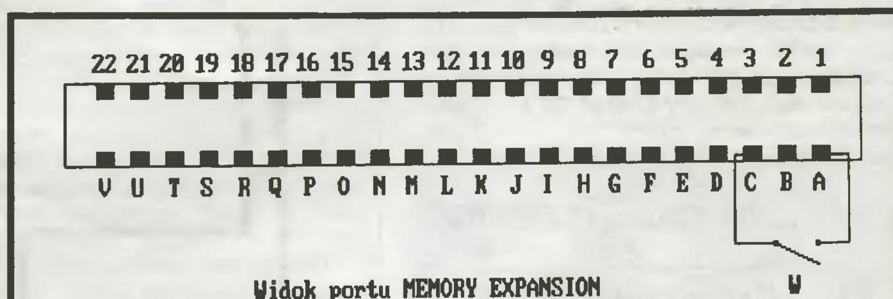


Widok gniazda SERIAL

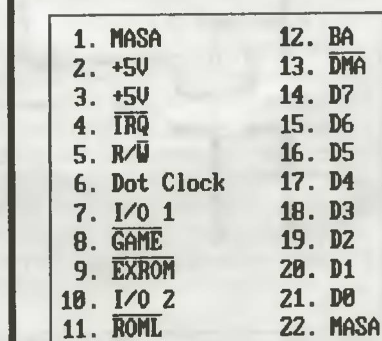


Widok gniazda krawędziowego USER PORT

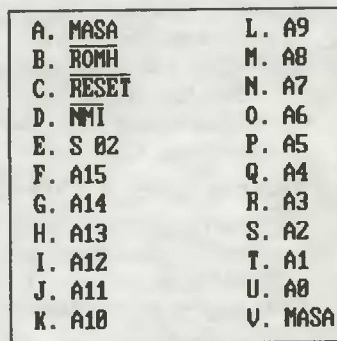
1. MASA	A. MASA
2. +5V	B. FLAG2
3. RESET	C. PB0
4. CNT1	D. PB1
5. SP1	E. PB2
6. CNT2	F. PB3
7. SP2	G. PB4
8. PC2	H. PB5
9. SERIAL ATN WE	I. PB6
10. 9V ~	J. PB7
11. 9V ~	K. PA2
12. MASA	L. MASA



Widok portu MEMORY EXPANSION



1. MASA	12. BA
2. +5V	13. DMA
3. +5V	14. D7
4. IRQ	15. D6
5. R/W	16. D5
6. Dot Clock	17. D4
7. I/O 1	18. D3
8. GAME	19. D2
9. EXROM	20. D1
10. I/O 2	21. D0
11. ROML	22. MASA



A. MASA	L. A9
B. ROMH	M. A8
C. RESET	N. A7
D. NMI	O. A6
E. S 02	P. A5
F. A15	Q. A4
G. A14	R. A3
H. A13	S. A2
I. A12	T. A1
J. A11	U. A0
K. A10	V. MASA

SKĄD SIĘ BIORĄ BŁĘDY

Pozornie może się wydawać, że na to pytanie jest bardzo łatwo odpowiedzieć. Niestety lektura redakcyjnej korespondencji, a także liczne zapytania oraz wątpliwości wskazują, że początkujący użytkownicy mają z błędami wiele problemów. W artykule tym chciałbym przedstawić kilka typowych, lecz czasami dość uciążliwych do wyłapania błędów.

Swoje wywody zacznę od zwalenia winy za takie pomyłki na użytkownika komputera. Dla niektórych zabrzmi to prawdopodobnie bardzo niemile, inni poczują się urażeni. „Jak to ? JA miałbym się pomylić ? Też coś ! To maszyna, głupie elektroniczne pudło się pomyliło, albo listing w czasopiśmie był źle wydrukowany !" — tak pomyśli sobie teraz wielu z was. Pogląd ten jest po pierwsze bardzo charakterystyczny (ludzie uwielbiają uważać się za nieomylnych), po drugie zaś — niezwykle popularny.

Czasami otrzymujemy od Was bardzo bezkrytyczne listy, takie jak np. „Chciałem wpisać program... ze strony... Po wpisaniu i uruchomieniu programu komputer zasygnalizował błąd... Sprawdzałem 82 razy z całą rodziną i błędu NIE popełniłem na pewno... Dlaczego drukujecie błędne listingi?" Zanim jednak zadasz takie pytanie zastanów się, czy aby na pewno nie popełniłeś błędu ukrytego. Jestem w stanie wykazać, że błędy mogą być bardzo trudno zauważalne, a komunikaty je opisujące są czasami bardzo mylące. Wpisz następujący programik:

```
100 FOR X = 1 TO 5
110 READ A
120 PRINT X, A
130 NEXT X
140 PRINT„GOTOWE !":END
150 :
160 DATA 001, 002, 003, 004, 005
```

Po uruchomieniu spowoduje on wyświetlenie na ekranie wartości zmiennej X i odczytywanych z linii DATA danych. Spróbujmy teraz „pomylić się". W tym celu zastąp przecinek pomiędzy liczbami 003 i 004 kropką, a następnie uruchom program ponownie:

```
160 DATA 001, 002, 003. 004, 005
```

Efekt jest dość nieoczekiwany; komputer bez zmrużenia oka przyjął błąd, wyświetlił FAŁSZYWE dane i zasygnalizował błąd w wierszu... 110 zamiast w 160 !

Zwróć uwagę na trzy sprawy: po pierwsze klawisz przecinka leży bardzo blisko klawisza kropki, po drugie wizualnie na ekranie oba znaki są do siebie dość podobne (trzeba się dobrze przyjrzeć, żeby je odróżnić), wreszcie po trzecie w długim ciągu danych taka „nieścisłość" jest naprawdę trudna do zauważenia.

OUT OF DATA to komunikat sygnalizujący niezgodność pomiędzy liczbą danych fizycznie zawartych w programie (w liniach z instrukcją DATA) a liczbą danych nakazanych do odczytania. W naszym przykładzie nieumyślne zastąpienie przecinka (ODDZIELAJĄCEGO DANE) kropką spowodowało w rezultacie połączenie dwóch danych (w liczbę 3.004). Zwróć uwagę, że jednocześnie skraca to liczbę danych do odczytania — są już teraz tylko cztery zamiast pięciu. Po przypisaniu zmiennej A liczby 3.004 komputer odczyta liczbę 5 (czyli OSTATNIĄ daną) w czwartym (PRZEDOSTATNIM) obiegu pętli. Próba odczytu kolejnej piątej wartości za pomocą instrukcji READ w linii 110 spowoduje zasygnalizowanie błędu, gdyż nie ma już więcej danych do odczytania. Ważne jest jednak, że błąd wystąpił w trakcie ODCZYTYWANIA danych i stąd komunikat mówi o linii 110 w której znajduje się instrukcja odczytująca dane. Proste ?

Drugim typowym błędem jest omyłka w cyferkach. Wprowadź teraz do programu dwa nowe wiersze:

```
120 POKE 5000+X,A
160 DATA 251, 252, 263, 254, 255
```

Podczas wpisywania zwróć szczególną uwagę na liczbę 263 zawartą w wierszu 160. Po uruchomieniu programu pojawi się dość dziwny komunikat: ILLEGAL QUANTITY ERROR. Jest on oczywiście związany z liczbą 263, ponieważ do dowolnej komórki pamięci C-64/128 czy C-16 można wpisać liczbę mniejszą lub równą 255.

Czasami możliwe jest określenie w której linii zawierającej instrukcję DATA wystąpił błąd. Po zasygnalizowaniu błędu wykonywanie programu jest zatrzymywane; wtedy wykonaj:

```
PRINT PEEK (63) *256 + PEEK(64) <RETURN>
```

W komórkach tych system operacyjny przechowuje adres ostatnio odczytywanej linii DATA; w ten sposób można czasami ustalić w którym miejscu wpisującemu powinęła się noga...

Do równie częstych i trudnych do wychwycenia błędów należą problemy z zerem. Wpisz poniższy program i uruchom go.

```
100 A0 = 06 : REM ZMIENNA "A ZERO"
110 O2 = 08 : REM ZMIENNA "LITERA O DWA"
120 B5 = 02 : REM WARTOŚĆ ZERO DWA
130 :
140 PRINT A0, 02, B5
```

Po uruchomieniu tego programu i sprawdzeniu jego działania zapisz sobie wartości wyświetlane w linii 140. Następnie zastąp liczbę 02 w wierszu 120 wyrażeniem 02 (litera 0 i 2) i uruchom program ponownie. Widać zresztą od razu co nastąpi: zmiennej B5 zostanie przyporządkowana liczba 8 zamiast 2 i cała reszta programu poda dalej fałszywe wyniki. Oczywiście lokalizacja takiej pomyłki w programie przykładowym nie sprawi nikomu żadnego kłopotu, spróbuj jednak znaleźć taki błąd w programie liczącym kilkaset linii!

Pamiętaj zatem: ze względu na podobieństwo litery O i cyfry 0 oraz na ich sąsiedztwo na klawiaturze staraj się nie używać żadnych zmiennych zawierających w nazwie jeden z tych znaków. Osobom o słabszym wzroku odradzam również stosowanie cyfry 8 i litery B.

Jednym z najbardziej niebezpiecznych błędów jest dokładność komputera. Problem bierze się stąd, że PRZED wykonaniem działania liczba całkowita jest zamieniana na tzw. wykładnik i mantysę; zamiana ta obniża dokładność obliczenia. W sytuacji, gdy użytkownik będzie usiłował wykonać działanie dające w rezultacie liczbę większą niż 1.70141183E+38 Commodore 64 wyświetli komunikat OVERFLOW ERROR. Niestety przekroczenie dolnej granicy, czyli liczby 2.93873588E-39 NIE JEST SYGNALIZOWANE NICZYM (w rezultacie otrzymasz w wyniku liczbę 0).

Choćbyś nie wiem jak chciał (lub chciała) przed błędami się nie ustrzeżesz. Od Twojego zaawansowania i doświadczenia będzie zależało jak szybko je odnajdziesz i usuniesz. Pomyłki w programach typowo obliczeniowych można np. wyszukiwać wykonując działania krok po kroku i wyświetlając na ekranie wartości zmiennych stosowanych do obliczeń tymczasowych i pośrednich. W moim wypadku sposób ten niejednokrotnie skrócił „odpluskwianie" programu z godzin do minut.

Jeśli jesteś początkujący możesz również dać się złapać na błąd w numeracji linii programu. A oto przykład (wpisuj ten program dokładnie tak jak to podano):

```
110 B = 10
115 C = 5
120 A = B*C
125 PRINT "WYNIK DZIAŁANIA";B;"*";C; "JEST
ROWNY";A
120 A = B/C
135 PRINT "WYNIK DZIAŁANIA";B;"/";C; "JEST
ROWNY";A
```

Jeżeli program ten będzie wpisywany w „normalny" sposób (czyli automatycznie, bez analizy tego co się właściwie wpisuje) to błąd ten musi wystąpić. Pomyłka w tym wypadku leży zwykle po stronie czasopisma i osoby przygotowującej listing do druku, chociaż roztargniony programista może ją zrobić tak samo łatwo. Zwróć uwagę, że po wpisaniu linii 125 wprowadzasz PONOWNIE linię 120 (zamiast planowanej 130), co interpreter BASIC traktuje jako jej zastąpienie (poprzedni wiersz oznaczony numerem 120 jest z pamięci usuwany). W efekcie oba wyrażenia w liniach 125 i 135 będą fałszywe!

Z naszej strony każdy sygnał o błędzie w programach publikowanych w „C&A" sprawdzamy bardzo uważnie. Nie znaczy to wcale, że nie zdarzają się nam pomyłki (patrz rubryka „CHOCHLIK GÓRĄ"). Jednakże doświadczenie wskazuje, że na linii komputer — człowiek znacznie bardziej zawodnym urządzeniem jest ten drugi i stąd moja prośba o nieco więcej samokrytyki dla siebie.

KLAUDIUSZ DYBOWSKI

NAJPROSTSZA BAZA DANYCH dla C-128

Nie muszę chyba nikogo przekonywać o ogromnej przewadze komputerowych baz danych nad tradycyjnymi metodami przechowywania informacji. Po pierwsze, jeden twardy dysk lub kilka dyskietek mogą pomieścić tyle informacfji, ile dawniej z trudem mieściło się w sporej szafie wypełnionej po brzegi papierem. Po drugie, wyszukanie konkretnej informacji spośród tysięcy innych zapisanych na dyskietce trwa zwykle kilka sekund (ile trwa ta sama operacja w tradycyjnym archiwum wie każdy, kto próbował coś załatwić w jednym z naszych wspaniałych, kultywujących stare zwyczaje urzędów). Wreszcie po trzecie, informacja raz zapisana na nośniku magnetycznym może być w razie potrzeby dowolnie modyfikowana, uaktualniana, podczas gdy informacje na papierze są martwe, wprowadzenie do nich jakiejś zmiany równa się najczęściej pisanlu wszystkiego od nowa.

No dobrze, myślę, że ten wstęp wzbudził u Was instynkt posiadania i na pewno natychmiast chcielibyście założyć sobie bazę danych. Proszę bardzo, wystarczy wpisać i uruchomić program z listingu 1.

LISTING 1

```
 90 rem listing #1
 99 :
100 rem * otwarcie zbioru REL "ad
    resy" *
101 rem *  dlugosc rekordu: 80 zn
    akow  *
102 :
110 p$=chr$(13):rv$=chr$(18)
111 scnclr
115 print rv$;"Nazwisko, imie:"
120 input a$
125 print rv$;"Ulica, nr domu/nr
    mieszkania:"
130 input b$
135 print rv$(18)"Kod, miasto:"
140 input c$
145 dopen#5,"adresy",l80:rem prze
    d 80 jest litera "el"
150 record#5,2,1
155 print#5,a$;p$;b$;p$;c$
160 dclose#5
165 la$=str$(2)
170 dopen#5,"adresy"
175 record#5,1,1
180 print#5,la$
185 dclose#5
190 scnclr:end
```

Program ten otwiera na dyskietce zbiór typu REL o nazwie „adresy", bo nasza baza danych będzie pełniła rolę książki adresowej. Spośród czterech typów zbiorów (PRG, SEQ, USR, REL), jakie ośmiobitowe komputery firmy Commodore mogą tworzyć na dyskietce, właśnie zbiór typu REL (relative) przeznaczony jest na potrzeby baz danych. Każdy zapis w takim zbiorze nosi nazwę rekordu i ma przyporządkowany oddzielny numer. Dzięki temu znalezienie danego rekordu przez system operacyjny stacji dysków odbywa się bardzo szybko, praktycznie w ciągu ułamka sekundy.

Przypatrzcie się dobrze linii 145. Polecenie DOPEN# służy do otwarcia kanału łączności między komputerem a stacją dysków. Numer kanału (w programie — 5) może być dowolny z zakresu 1–14 (na C-128 istnieje 15 kanałów, ale ostatni zarezerwowany jest dla systemu operacyjnego). Pamiętajcie, że raz wybrany numer kanału obowiązywać będzie zawsze w danym zbiorze. Dotyczy to nie tylko instrukcji DOPEN#, ale także INPUT#, GET#, PRINT#, RECORD# i DCLOSE#.

W omawianej linii programu po nazwie zbioru występuje jeszcze tajemnicze wyrażenie „L80". Określa ono, ile znaków (maksymalnie) będzie zawierał każdy rekord. Nic się nie stanie, jeśli wpisywany przez nas rekord będzie miał mniej niż przyjęte tu 80 znaków. Ale gdy będzie miał więcej, DOS „obetnie" wszystkie nadwyżkowe znaki, może też zasygnalizować błąd „OVERFLOW IN RECORD" (przepełnienie w rekordzie). Na C-128 możemy ustalać długość rekordów w zakresie od 2 do 254 znaków, a ustalenia takiego dokonujemy tylko raz — w trakcie tworzenia bazy danych. Roztargnionym przypominam, że literka „L" przed liczbą określającą długość rekordu jest niezbędna!

Przejdźmy teraz do linii 150. Instrukcja RECORD# umożliwia zapis lub odczyt danych z rekordu o numerze określonym pierwszą liczbą po przecinku. Zdziwicie się pewnie, dlaczego zaczynamy nasz zbiór od drugiego rekordu? Dlatego, że w pierwszym zapisywać się będzie automatycznie liczba rekordów zawartych w naszej książce adresowej (są za to odpowiedzialne linie 180–200 z listingu nr 2). Liczba ta będzie przecież z czasem rosła, a byłoby cokolwiek śmieszne, gdybyśmy posiadając komputerową bazę danych, zapisywali na kartce papieru, ile właściwie mamy rekordów. No tak, spytacie, ale po co zapamiętywać liczbę rekordów? Po to, żeby przez pomyłkę nie skasować zapisów już istniejących. Skąd będę wiedział, jaki numer rekordu nadać adresowi nowego znajomego, jeżeli ostatni adres wpisywałem rok temu?

Ostatni parametr instrukcji RECORD# określa, od którego znaku komputer ma odczytać bądź zapisać dane w rekordzie. Możliwości wynikające z istnienia tego parame-

tru są ogromne, żeby je dokładnie opisać, trzeba by poświęcić im oddzielny artykuł. Dlatego w naszych rozważaniach ograniczymy się do operowania rekordami w całości, od pierwszego znaku.

Działanie linii 155 jest chyba dla wszystkich jasne: instrukcja PRINT# zapisuje w rekordzie nazwisko i adres w postaci wcześniej zdefiniowanych zmiennych. Kod CHR#(13) (tu jako zmienna P$ — patrz linia 110) nakazuje komputerowi zapisać dane od nowego wiersza (tak, jakbyśmy nacisnęli klawisz RETURN). Nierozdzielenie w ten sposób danych podczas zapisu sprawia, że rekord po odczytaniu jest nieczytelny — znaki tworzą jeden ciąg. Zwróć uwagę, że kod CHR$(13) traktowany jest w rekordzie jak każdy inny znak, tj. wlicza się w długość rekordu (zamiast ustalonych 80 znaków, będziemy więc mieli tylko 78 efektywnych).

I wreszcie linia 160 zamyka kanał łączności ze stacją dysków, a linie 165–185 wpisują do pierwszego rekordu liczbę 2 — bo tyle właśnie rekordów liczy sobie na razie nasz zbiór.

LISTING 2

```
 90 rem listing #2
 99 :
100 rem * zapis adresow w zbiorze
    REL *
101 :
103 p$=chr$(13):rv$=chr$(18)
104 scnclr
105 dopen#5,"adresy"
110 record#5,1,1
115 input#5,la$
120 dclose#5
125 la=val(la$):nr=la+1
130 print rv$;"Nazwisko, imie:"
135 input a$
140 print rv$;"Ulica, nr domu/nr
    mieszkania:"
145 input b$
150 print rv$;"Kod, miasto:"
155 input c$
160 dopen#5,"adresy"
165 record#5,(nr),1
170 print#5,a$;p$;b$;p$;c$
175 dclose#5
180 la$=str$(nr)
185 dopen#5,"adresy"
190 record#5,1,1
195 print#5,la$
200 dclose#5
205 print:print"Koniec ? (t/n)"
210 getkey k$
215 if k$="t" then scnclr:end
220 goto 100
```

Mamy więc założoną bazę danych. Teraz potrzebne nam będą programy obsługujące.

Pierwszy z nich (listing 2) przeznaczony jest do wpisywania nowych adresów. Po jego uruchomieniu wystarczy jedynie podawać dane, o które pyta komputer. Nadanie rekordom kolejnych numerów, wypełnianie ich danymi i powiększenie o jeden liczby zawartej w pierwszym rekordzie odbywa się automatycznie. Jeżeli wpiszecie nie jeden, lecz pięć czy dziesięć nowych adresów, będziecie mogli zobaczyć, jak powiększyła się nasza baza danych: powinna teraz zajmować cztery lub więcej bloków (poprzednio dwa).

LISTING 3

```
 90 rem listing #3
 99 :
100 rem * odczyt wg nazwiska *
101 :
104 scnclr:ro=0:rv$=chr$(18)
105 print rv$;"Podaj nazwisko:"
110 input a$:scnclr
115 print rv$;"Szukane nazwisko:
    "na$
120 print
125 dn=len(na$):p=2
130 dopen#5,"adresy"
135 record#5,(p),1
140 gosub 180
145 input#5,a$,b$,c$
150 if left$(a$,dn)=na$ then 160
155 p=p+1:goto 135
160 ro=ro+1
165 print rv$;"Rekord nr"p
170 print a$:print b$:print c$:pr
    int
175 p=p+1:goto 135
180 if ds<>50 then return
185 dclose#5
190 if ro=0 then print rv$;"Nazwi
    ska nie znaleziono.":end
```

Program z listingu 3 wyszukuje jedno podane przez nas nazwisko, a następnie wyświetla je wraz z imieniem, pełnym adresem i numerem rekordu. Jeżeli nie pamiętacie całego nazwiska, wystarczy, że podacie tylko pierwszą literę — komputer wyświetli wtedy wszystkie nazwiska zaczynające się na tą literę. Program przeszukuje każdorazowo całą bazę danych na wypadek, gdybyście mieli kilku znajomych o tym samym nazwisku. Na pewno chcielibyście jeszcze wiedzieć, skąd program „wie", że sprawdził już wszystkie rekordy? Odczytuje po prostu wartość zmiennej systemowej DS (disk status) w linii 180. W zmiennej tej przechowywane są numery komunikatów systemu operacyjnego stacji dysków. Komunikat, o który nam chodzi, ma numer 50 i brzmi „RECORD NOT PRESENT" (brak rekordu).

Ostatnie dwa programy nie wymagają komentarza, poza tym, że tempo wyświetlania rekordów w programie nr 5 można regulować pętlą opóźniającą zawartą w linii 135.

Na zakończenie mam dla Was propozycję: spróbujcie napisać samodzielnie program umożliwiający zapis, odczyt i wydruk rekordów w porządku alfabetycznym. Wysiłek z pewnością się opłaci.

CHRISTIAN GRZENKOWICZ

LISTING 4

```
 90 rem listing #4
 99 :
100 rem * odczyt rekordow *
101 :
100 scnclr:rv$=chr$(18)
105 print rv$;"Podaj numer rekord
    u:"
110 input nr:scnclr
115 print
120 dopen#5,"adresy"
125 record#5,(nr),1
130 gosub 175
135 input#5,a$,b$,c$
140 print rv$;"Rekord nr"nr
145 print a$:print b$:print c$
150 dclose#5
155 print:print "Koniec ? (t/n)
160 getkey k$
165 if k$="t" then scnclr:end
170 goto 100
175 if ds<>50 then return
180 dclose#5
185 print rv$;"Nie ma takiego rek
    ordu.":goto 155
```

LISTING 5

```
 90 rem listing #5
 99 :
100 rem * odczyt wszystkich rekor
    dow zbioru *
102 :
104 scnclr:nr=1:rv$=chr$(18)
105 dopen#5,"adresy"
110 record#5,(nr),1
115 gosub 145
120 input#5,a$,b$,c$
125 print rv$;"Rekord nr"nr
130 print a$:print b$:print c$:pr
    int
135 for po=0 to 250:next
140 nr=nr+1:goto 110
145 if ds<>50 then return
150 dclose#5
155 print rv$;"Koniec.":end
```

TURBO dla C-16/116/PLUS/4

Miesiąc temu pisałem o kasetowych systemach TURBO dla rodziny C-16. Jeśli nie pamiętasz o czym była tam mowa zajrzyj do poprzedniego numeru, a następnie przyjrzyj się rysunkowi. Jest on wynikiem następującego testu: grę „EXORCIST" o długości mniej więcej 12 kB (to jest 49 bloków na dysku) zapisałem w czterech opisywanych systemach TURBO oraz — dla porównania — bez TURBO i na dyskietce (stacja 1551). Następnie ze stoperem w ręku mierzyłem czas odczytu.

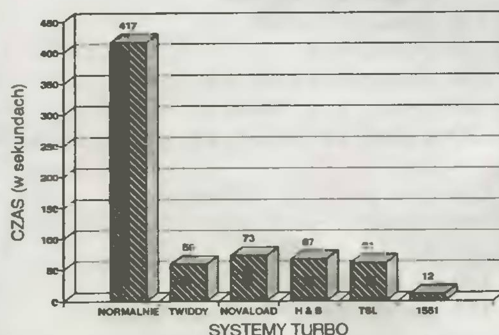
Mam nadzieję, że jasno widać teraz sens zapisu programów w TURBO — przyspieszenie wczytywania w stosunku do normalnego zapisu jest aż siedmiokrotne. Czas oczekiwania na wczytywanie ulubionej gry możemy więc skrócić z siedmiu minut do jednej, a poza tym oszczędzamy miejsce na kasecie. Widać także, o ile krócej czekają szczęśliwi posiadacze stacji 1551. W wypadku stacji 1541 wczytywanie programów zajmuje niemal tyle samo czasu, co z magnetofonu w systemie TURBO! Kończąc ten temat sugerowałbym stosowanie dwóch systemów: TSL do zapisu własnej twórczości oraz NOVALOAD do zapisu gier.

Fakt, że spotykamy się na łamach „C&A" po raz kolejny upoważnia mnie do pisania o nieco ogólniejszych sprawach. Chciałbym Was niniejszym poprosić o nadsyłanie listów z propozycjami artykułów o C-16/116/PLUS/4, które chcielibyście znaleźć w naszym piśmie.

Z własnego doświadczenia wiem, że spora grupa użytkowników nie ma dostępu do żadnej literatury, nawet do tłumaczenia oryginalnej instrukcji obsługi. Stąd niektórzy z was nie dają sobie prawdopodobnie rady nawet z najprostszymi problemami. Przypominam więc, że zawsze możecie zwracać się z nimi do mnie listownie bądź telefonicznie (piątki o godzinie 17.00).

WOJCIECH KAZIMIERCZAK

SYSTEMY TURBO DLA RODZINY COMMODORE 16/116/PLUS/4 - POROWNANIE



DEFINIOWANIE POLSKICH LITER na DRUKARKACH STAR

Przedstawiony poniżej program umożliwia zdefiniowanie i korzystanie z polskich liter na drukarkach firmy Star z rodziny NX oraz LC. Polskie litery uzyskano poprzez zastąpienie rzadziej używanych znaków dostępnych na Commodore 64.

Drukarki te mają możliwość przeniesienia generatora znaków z pamięci ROM do RAM (downloading). Wydruk polskich liter nie obejmuje trybu pracy NLQ — znaki zaprogramowano dla trybu pracy DRAFT drukarki.

Przedstawiony program może stanowić przykład definiowania własnych znaków graficznych na innych typach drukarek, które mają możliwości przenoszenia generatora znaków z pamięci ROM do RAM.

Po włączeniu drukarki i komputera należy wczytać i uruchomić program i postępować zgodnie z instrukcjami wyświetlanymi na ekranie. Po wykonaniu programu komputer zostaje wyzerowany instrukcją SYS 64738 (linia 500 — RESET) i od tej chwili znaki wysyłane do drukarki (takie jak w linii 510) są drukowane jako polskie litery. Znaki te są dostępne zarówno podczas pracy w BASIC, jak i pod kontrolą edytora tekstu, np. EASYSCRIPT.

ANDRZEJ KĘPA

```
 98 rem *** download nx-10 ***
 99:
100 print chr$(147);chr$(14);spc(12)
    ;"Polskie znaki"
105 print tab(7)"Prosze uruchomic dr
    ukarke"
110 print tab(7)"i wcisnac dowolny k
    lawisz"
115 get a$:if a$="" then 115
120 print tab(7)"Czy wydrukowac tabl
    ice"
125 print tab(7)"zamiennikow (t/n)";
130 input a$
135 open 4,4,5
140 print#4,chr$(27);chr$(64);:print
    #4,chr$(27);"g";
145 if a$<>"t" then 155
150 print#4,spc(10);"ekran";:gosub 2
    30
155 print#4,chr$(27);":";chr$(0);chr
    $(0);chr$(0);
160 c$=chr$(27)+chr$(38)+chr$(0)
165 for j=1 to 10:b$=c$
170 for i=1 to 14:read c:b$=b$+chr$(
    c):next
175 print#4,b$;:next
180 print#4,chr$(27);"%1";chr$(0);
185 close 4
190 if a$<>"t" then 215
195 open 4,4,7
200 print#4
205 print#4,spc(7);"DRUKARKA";:gosub
    230
210 close 4
215 print"Polskie znaki zainstalowan
    e"
220 for x=1 to 1000:next
225 sys 64738
230 print#4,spc(10);" # $ % & ' [ ]
    < >"
235 return
240 :
245 data 035,035,011,008,020,064,020
    ,064,020,066,057,004,000,000
250 data 036,036,139,028,034,000,034
    ,064,162,000,034,000,000,000
255 data 037,037,011,056,068,016,068
    ,016,070,017,068,048,000,000
260 data 038,038,139,000,138,000,254
    ,000,034,000,000,000,000,000
265 data 039,039,139,254,000,018,000
    ,034,000,066,000,002,000,000
270 data 064,064,139,062,000,032,000
    ,096,128,032,030,000,000,000
275 data 091,091,139,028,034,000,034
    ,064,162,000,034,028,000,000
280 data 093,093,139,016,042,000,042
    ,064,170,000,042,004,000,000
285 data 060,060,139,034,004,034,136
    ,034,016,034,000,000,000,000
290 data 062,062,139,034,004,034,072
    ,162,016,034,000,000,000,000
```

6526 COMPLEX INTERFACE ADAPTER # 1

część 1

Twój C-64 (i nie tylko — C-128 zawiera również te kości) byłby niewiele wart bez tych układów. Układy te, potocznie nazywane „portami" służą właściwie wyłącznie do obsługi urządzeń zewnętrznych zwanych również „urządzeniami we/wy".

Urządzenia we/wy (wejścia/wyjścia) to klawiatura, stacja dysków, magnetofon, joystick, paddle, pióro świetlne, monitor, drukarka, plotter — jednym słowem wszystko, co da się przyłączyć do Twojego komputera.

Układ CIA #1 spełnia wiele funkcji: obsługuje klawiaturę, joysticki i wiosełka (paddles), zarządza przerwaniami IRQ i NMI. CIA #1 obsługuje także transmisję szeregową poprzez port użytkownika (USER PORT). Układ ten ma również wbudowany zegar czasu rzeczywistego oraz dwa timery.

Omówię teraz cztery pierwsze rejestry CIA#1. Przedstawiam je razem, ponieważ ich działanie jest ściśle ze sobą połączone.

56320 ($DC00) CIAPRA CIA#1+0

Port danych A. Ma różne funkcje w zależności od przeprowadzanej operacji (w sensie zapisu lub odczytu). Obsługuje klawiaturę, joystick #2, przyciski FIRE wiosełek.
Na rysunku 1 przedstawiono matrycę klawiatury. Aby wybrac do odczytu określoną kolumnę z matrycy klawiatury należy odpowiedni bit CIAPRA wyzerować.

bit 0 : zapis — wybór odczytu kolumny 0 matrycy klawiatury,
: odczyt — stan „góra" joysticka #2;
bit 1 : zapis — wybór odczytu kolumny 1 matrycy klawiatury,
: odczyt — stan „dół" joysticka #2;
bit 2 : zapis — wybór odczytu kolumny 2 matrycy klawiatury,
: odczyt — stan „lewo" joysticka #2,
: odczyt — stan „fire" paddle #1
bit 3 : zapis — wybor odczytu kolumny 3 matrycy klawiatury,
: odczyt — stan „prawo" joysticka #2,
: odczyt — stan „fire" paddle #2;
bit 4 : zapis — wybór odczytu kolumny 4 matrycy klawiatury,
: odczyt — stan „fire" joysticka #2;
bit 5 : zapis — wybór odczytu kolumny 5 matrycy klawiatury,
bit 6 : zapis — wybór odczytu kolumny 6 matrycy klawiatury,
: zapis — wybór odczytu wiosełek z portu A lub B;
bit 7 : zapis — wybór odczytu kolumny 7 matrycy klawiatury,
: zapis — wybor odczytu wiosełek z portu A lub B.

56321 ($DC01) CIAPRB CIA#1+1

Port danych B. Jego głównym zadaniem jest odczyt stanu klawiatury oraz joysticka #1. Tutaj też można odczytać stan przycisków fire wiosełek nr 3 i 4. Ponadto bity 6 i 7 służą jako wyjście dla timerów A i B. Dla klawiatury i joysticków przyjęto, że stan zero oznacza odpowiednio naciśnięty klawisz lub zmianę położenia joysticka.

bit 0 : odczyt — stan rzędu 0 matrycy klawiatury,
: odczyt — stan „góra" joysticka #1;
bit 1 : odczyt — stan rzędu 1 matrycy klawiatury,
: odczyt — stan „dół" joysticka #1;
bit 2 : odczyt — stan rzędu 2 matrycy klawiatury,
: odczyt — stan „lewo" joysticka #1,
: odczyt — stan „fire" paddle #3;
bit 3 : odczyt — stan rzędu 3 matrycy klawiatury,
: odczyt — stan „prawo" joysticka #1,
: odczyt — stan „fire" paddle #4;
bit 4 : odczyt — stan rzędu 4 matrycy klawiatury,
: odczyt — stan „fire" joysticka #1;
bit 5 : odczyt — stan rzędu 5 matrycy klawiatury;
bit 6 : odczyt — stan rzędu 6 matrycy klawiatury,
: odczyt — linia wyjściowa timera A;
bit 7 : odczyt — stan rzędu 7 matrycy klawiatury,
: odczyt — linia wyjściowa timera B;

56322 ($DC02) CIADRA CIA#1+2

Rejestr kierunku danych dla portu A. Służy do sterowania (i określenia) kierunku przepływu danych przechodzących przez port A. I tu mała uwaga. Przez **wejście** rozumiemy odczyt sygnałów z zewnątrz (odczyt danych z rejestru CIAPRA), przez wyjście wysyłanie sygnałów na zewnątrz (zapis do rejestru CIAPRA). Kolejne bity rejestru CIADRA odpowiadają bitom rejestru CIAPRA. Standardowo do CIADRA wpisywana jest wartość 255 ($FF) — wszystkie linie portu danych A są więc zdefiniowane jako wyjście. Jeżeli wybrany bit rejestru CIADRA jest wyzerowany, to odpowiadający mu bit rejestru CIAPRA jest traktowany jako wejście (zapis), w przeciwnym wypadku — jako wyjście (odczyt).

56323 ($DC03) CIADRB CIA#1+3

Rejestr kierunku danych dla portu B. Kieruje przepływem danych przez port danych B. Standardowo wpisywana jest tu wartość 0 (wszystkie linie rejestru CIAPRB są ustawione jako „wejście"). Działa tak samo jak rejestr CIADRA, ale w połączeniu z rejestrem CIAPRB. Czas na opis i przykłady.

KLAWIATURA

Głównym zadaniem czterech pierwszych rejestrów CIA#1 jest obsługa klawiatury. To właśnie za pomocą rejestrów CIAPRA i CIAPRB C-64 „czyta" klawiaturę. Dzieje się to w dość specyficzny sposób. Na początku należy wpisać wartość 255 ($FF) do rejestru CIADRA, co ustawi wszystkie bity rejestru CIAPRA jako wyjście (zapis), oraz przesłać wartość 0 do rejestru CIADRB — ustawia to wszystkie bity portu danych B jako wejście (odczyt). Teraz spójrz na matrycę klawiatury. Należy wybrać wiersz matrycy do sprawdzenia. Niech będzie to wiersz odpowiadający bitowi 0. W tym celu należy wpisać wartość 254 ($FE) do rejestru CIAPRA. Na początku w tym rejestrze znajduje się wartość 255 ($FF) — żaden wiersz nie został „wybrany". Aby „wybrać" testowany wiersz trzeba wyzerować odpowiadający mu bit. Jeśli więc zostanie wyzerowany bit 0 w CIAPRA będzie to oznaczać, że został wybrany wiersz 0.

Po dokonaniu wyboru wiersza wystarczy już tylko poczekać do chwili naciśnięcia któregoś z klawiszy (leżących rzecz jasna w tym wierszu). I jeśli któryś z bitów w rejes trze CIAPRB zostanie wyzerowany — oznacza to, że klawisz został naciśnięty. Na przykład: wyzerowany bit 0 tego rejestru oznacza naciśnięcie klawisza DELETE. Zasadę działania obrazują również programy 1 i 2 — oba oczekują na wciśnięcie tego klawisza. Łatwo też zauważyć, że program 2 jest dużo mniej przejrzysty od programu 1.

Powróćmy jeszcze raz do matrycy klawiatury. Brakuje tutaj klawisza RESTORE. Jest on dołączony bezpośrednio do linii NMI procesora i za każdym jego naciśnięciem wywoływane jest przerwanie NMI — CIA#1 nie ma tu nic do gadania. Natomiast klawisz SHIFT LOCK jest dokładnie tym samym co lewy SHIFT. Ot, binarna logika — 8 wierszy i 8 kolumn i już mamy 64 klawisze + 2 dodatkowe.

Procedura przerwań IRQ dokonuje odczytu klawiatury co 1/60 (w Europie 1/50) sekundy. Jeśli nawet zablokujemy przerwania, to można zawsze skorzystać z procedury SCNKEY (KERNAL — $FF9F).

Po odczycie klawiatury klawiszom przyporządkowywane są kody z tablicy kodów klawiatury ($EB81–$EC43) wraz ze sprawdzeniem znaczników C=, SHIFT i CONTROL. Następnie kod znaku zostaje przesłany do bufora klawiatury ($200–$258) i wyświetlony (o ile C-64 jest w trybie bezpośrednim, lub INPUT). To jeszcze nie wszystko. Kodom klawiszy zostają podporządkowane formy graficzne pobierane z generatora znaków ($D000–$DFFF) lub specjalne funkcje i dopiero wtedy znak zostaje wyświetlony w postaci ciągu 64 bitów odpowiadających matrycy znaku.

Program 3 pokazuje jak można oszukać nasz komputer. Interpreter nie wie dokładnie, w którym wierszu został naciśnięty klawisz i stąd symbole ",","N","V","X",CRSRDOWN działają tak samo jak klawisz STOP (który dla interpretera ma specjalną funkcję).

JOYSTICKI

Na początek warto zauważyć, że joystick #2 jest dołączony do portu danych A, a joystick #1 do CIAPRB (logicznie powinno być na odwrót). Joystick może wysyłać do odpowiadającego mu portu (po uprzednim ustawieniu tego rejestru jako wyjście) pięć sygnałów: góra, lewo, prawo, dół, fire. Sygnały wysyłane przez joysticki można odczytać za pośrednictwem pięciu młodszych bitów rejestrów CIAPRA i CIAPRB. Tak samo, jak w wypadku klawiatury — stan 0 określonego bitu oznacza, że został wysłany sygnał z joysticka. Przykładowo joystick #1 (CIAPRB) można odczytać z poziomu BASIC instrukcją:

```
PRINT NOT PEEK(56321) AND 15
```

W ten sposób zostanie odczytana zawartość rejestru CIAPRB. Następnie odczytana liczba jest poddawana operacji logicznej AND z liczbą 15 (%0001111). Bity 0–3 nie zmieniają swojej wartości, pozostałe są zerowane. Operacja NOT zmienia wszystkie 0 na 1 i odwrotnie. Odczytywane wartości będą odpowiadać następującym stanom joysticka #1:

0 — nic %0000
1 — góra %0001
2 — dół %0010
4 — lewo %0100
8 — prawo %1000

Pozycje pośrednie:

5 — góra i lewo %0101
6 — dół i lewo %0110
9 — góra i prawo %1001
10 — dół i prawo %1010

Wartości 3, 7, 11, 12, 13, 14, 15 nie pojawiają się, bo niemożliwe jest takie ustawienie joysticka w którym jednocześnie sygnalizowany będzie stan „góra" i „dół". W podobny sposób można odczytać joystick #2 tym razem z rejestru CIAPRA:

```
PRINT NOT PEEK(56320) AND 15
```

Odczyt przycisku fire joysticka #1 odczytasz instrukcją:

```
PRINT NOT ((PEEK(56321) AND 16)/16)
```

gdzie stan logiczny 1 oznacza „naciśnięty", a 0 — „nienaciśnięty". Tak samo można odczytać położenie przycisku FIRE joysticka #2, tyle tylko, że z rejestru CIAPRA.

Rejestr CIAPRB jest używany zarówno od odczytu klawiatury, jak i joysticka #1 — czasami może to prowadzić do pomyłek. Procedura sprawdzająca klawiaturę nie rozróżnia, czy bit został wyzerowany poprzez naciśnięcie klawisza, czy też ruch joysticka. Na przykład stan „prawo" joysticka #1 jest interpretowany jako naciśnięcie klawisza „2" (pozostałe stany joysticka #1 odpowiadają klawiszom: CTRL, strzałka w lewo, 1, spacja — gdyż leżą w tym samym wierszu i są obsługiwane przez bit 7 rejestru CIAPRA). Można tego uniknąć wyłączając po prostu przerwania generowane przez timer A na czas odczytu joysticka (POKE 56333,127:POKE 56320,255); po odczycie stanu joysticka #1 trzeba ponownie włączyć przerwania (POKE 56333,129). Całą tę operację można przeprowadzić za pomocą bufora klawiatury zerując go po odczytaniu stanu joysticka #1 (POKE 198,0). Jeżeli chcesz zobaczyć jak wygląda rejestr CIAPRA w czasie odczytu joysticka #2 — uruchom program 4. Trzymając przycisk fire i zmieniając położenie joysticka możesz zobaczyć, które bity tego rejestru są zerowane przy określonych położeniach joysticka. Ruchy joysticka #2 można symulować na klawiaturze za pomocą kombinacji: SPACE+2, SPACE+C, SPACE+B, SPACE+F1, SPACE+M. Nie ma możliwości wyłączenia odczytu joysticka #1 i zinterpretowania jego ruchu jako naciśnięcia klawisza. Jest to główna przyczyna, dla której większość gier wykorzystuje joystick #2.

WIOSEŁKA

Ponieważ pominąłem odczyt wiosełek w opisie układu SID, opiszę je teraz. Ustawienie bitu 7 w rejestrze CIAPRA i wyzerowanie bitu 6 spowoduje odczyt wiosełek dołączonych do portu 1. Jeśli bity te zostaną ustawione ponownie, to odczytywane będą wiosełka z portu 2. Wiosełka są obsługiwane przez układ SID.

Przed wyborem pary odczytywanych wiosełek należy wyłączyć przerwania IRQ, za pomocą np. POKE 56333,127. Stan wiosełek odczytuje się z rejestrów SID : 54297 ($D419) — wiosełka 1 i 3, 54298 ($D41A) — wiosełka 2 i 4. Przyciski FIRE wiosełek 1 i 3 można odczytać odpowiednio na podstawie stanu bitu numer 2 rejestrów CIAPRA i CIAPRB (0 oznacza, że przycisk został naciśnięty). Przyciskom FIRE wiosełek 2 i 4 przypisane są bitom o numerze 3 w rejestrach CIAPRA i CIAPRB. Aby odczytać stan przycisków FIRE wiosełek wykonaj:

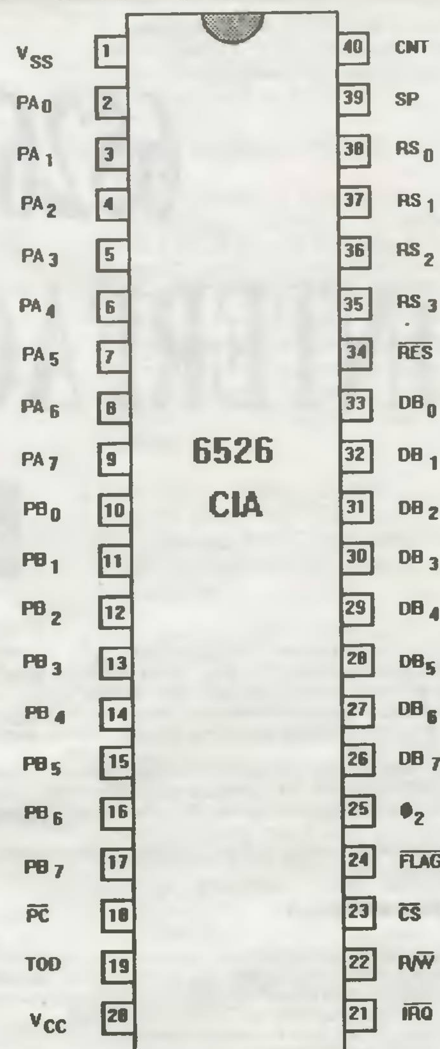
```
PD1 = NOT ((PEEK(56320)AND 4)/4)
PD2 = NOT ((PEEK(56320)AND 8)/8)
PD3 = NOT ((PEEK(56321)AND 4)/4)
PD4 = NOT ((PEEK(56321)AND 8)/8)
```

gdzie stan logiczny 1 oznacza przycisk wciśnięty.

Rejesr CIAPRB może być także wykorzystany jako linia wyjściowa dla timerów A i B. Przy odpowiednim ustawieniu rejestrów sterujących timerów A i B, po zliczeniu do zera zostaną odpowiednio ustawione bity 6 i 7.

Listingi przedstawiam na stronie 33 (cdn)

BAD



ODCZYT W PORCIE B (56321)

ZAPIS W PORCIE A (56320)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Bit 7	STOP	Q	C=	SPACJA	2	CTRL	←	1
Bit 6	/	↑	=	PRAWY SHIFT	HOME	;	*	£
Bit 5	,	@	:	.	–	L	P	+
Bit 4	N	O	K	M	0	J	I	9
Bit 3	V	U	H	B	8	G	Y	7
Bit 2	X	T	F	C	6	D	R	5
Bit 1	LEWY SHIFT	E	S	Z	4	A	W	3
Bit 0	KURSOR W DÓŁ	F5	F3	F1	F7	KURSOR W GÓRĘ	RETURN	DEL

Matryca klawiatury C-64

PAMIĘTNIK ARTYLERZYSTY:

JAK ZMUSIĆ KOMPUTER DO ROBIENIA RZECZY NIEMOŻLIWYCH

Wykład o „cyklowaniu", jaki sprawiłem Wam w zeszłym miesiącu nie był, co prawda, bardzo ekscytujący, ale istotny. Wiadomości z niego przydadzą się Wam nie raz, a jako przykład mogę podać choćby dzisiejszy, piąty już odcinek „pamiętnika znalezionego w armacie".

Sądząc z listów prowadzony na łamach „C&A" „Pamiętnik artylerzysty" się podoba. Od niektórych dostaję pochwały, od innych — baty, przychodzą również listy z pytaniami. W jednym z nich, napisanym przez członka jednoosobowej grupy The Monster przeczytałem, że człowiek ten ma zamiar przestać programować, jeśli nie podam mu sposobu na wyświetlanie duszków na obwódce ekranu. Apel był w takiej formie, że postanowiłem czym prędzej podzielić się tym sekretem z Czytelnikami „C&A".

Zacząć trzeba od tego, że firma Commodore Business Machines była przekonana, że nikt, nigdy i niczego nie wyświetli na obwódce ekranu, która po to zresztą powstała. Z tego przekonania (jak się okazało — błędnego) wywiodła ją wkrótce grupa, bodajże, 1001 Crew. Osłupiałym widzom zaprezentowała duszki, które tańczyły sobie po górnej i dolnej obwódce!

Tok rozumowania koderów przebiegał mniej więcej tak:

1. Jeżeli ustawimy (stan logiczny 1) bit 3 w komórce 53265 ($d011), to komputer wyświetli na ekranie 25 linii.
2. Jeżeli wyzerujesz bit 3 w tej komórce, to widoczne będą jedynie 24 linie, ostatnia linia schowa się zaś pod obwódką.
3. Wynika z tego, że komputer kończy wyświetlanie tła, a zaczyna obwódkę w zależności od stanu trzeciego bitu w komórce 53265, $d011.
4. Zróbmy więc mały eksperyment: nakażmy komputerowi wyświetlać 25 linii. Kiedy zacznie wyświetlać linię o numerze 25, zmień zdanie i nakaż mu wyświetlanie 24 linii.

Co, teoretycznie, powinno się stać? Po rozpoczęciu wyświetlania 25 linii komputer czeka do jej końca, by zacząć wyświetlanie obwódki. Jednak w trakcie tej czynności przenosimy wskaźnik końca tła na POCZĄTEK tej linii. A początek linii 25 już minęliśmy! No i co teraz? Sprawdź sam, najlepiej układając własny program, który spełni te założenia, albo po prostu, wpisz PROGRAM 1 i uruchom go instrukcją SYS 10000.

I co? Ciekawe? Tak jak chcieliśmy! Komputer zgłupiał i obwódki nie wyświetlił WCALE! Konstruktorzy VIC 6566 i Commodore 64 prawdopodobnie nawet nie przypuszczali, że taki numer jest możliwy. Zetknęliśmy się tutaj z pierwszą cechą układu wizji, której nie opisano w żadnej oficjalnej instrukcji obsługi; zapewniam was, że będzie ich więcej. Swoją drogą to ciekawe, że komputer działa nie tak jak powinien...

Zapewne część z Was będzie nieco rozczarowana po uruchomieniu tego programu. Mogą się bowiem pojawić czarne pionowe pasy. Nie jestem w stanie powiedzieć, na ilu maszynach będą widoczne. Z początku też nie potrafiłem sobie dać z tym rady. Potem jednak, albo przez jakieś pokrętne kombinowanie, albo metodą prób i błędów doszedłem do wniosku, że odpowiedzialna jest za to dziwne zjawisko komórka 16383, $3fff. Komórka ta jest ostatnim bajtem banku graficznego adresowanego aktualnie przez VIC. Jeżeli wpiszemy do niej 255 ($ff) — cała górna i dolna obwódka będzie czarna. Jeżeli wpiszemy 0 ($00) — obwódka będzie wyświetlana w kolorze tła. Wszelkie inne wartości dadzą węższe lub szersze, rzadsze lub gęstsze pionowe pasy. Przy odrobinie dobrej woli można je wykorzystać do czegoś pożytecznego.

Po tej krótkiej dygresji powróćmy wreszcie do długo oczekiwanych duszków. Wpiszcie przygotowany przeze mnie PROGRAM 2. Będzie w BASIC, bo BASIC jest w tym wypadku znacznie wygodniejszy — wcale nam na prędkości nie zależy.

Po uruchomieniu tego krótkiego programiku na ekranie pokazał się duszek, którego na konkurs piękności bym raczej nie wystawiał. Jeżeli naciśniesz jakikolwiek klawisz, duszek przesunie się o 1 punkt w dół. Za wartość początkową przyjąłem 0 ($00).

Od razu na początku — zaskoczenie! Duszek pokazał się na dole ekranu, a na górze widać tylko osiem jego linii. Jeżeli przytrzymamy klawisz jeszcze trochę, z dołu zniknie, wyjeżdżając poza ekran, na górze zaś pokaże się w całej okazałości. Dlaczego tak się dzieje? Nie wiem. Dorobiłem sobie do tego taką ideologię, że w poziomie mamy dodatkowy, dziewiąty bit położenia i dzięki temu na ekranie szerszym niż 255 punktów wszystko trzyma się kupy. W pionie mamy zaś (teoretycznie) 200 punktów, więc dodatkowy bit jest niepotrzebny. W praktyce jednak — jak widzimy — linii pionowych mamy więcej, więc komputer dla porządku wyświetla oba możliwe położenia duszka.

Przesuwajmy jednak dalej... Na moim monitorze (Neptun 156), po dojściu do 48 linii duszek był cały, dobrze widoczny, lecz — i tu następna niespodzianka — na samej górze pojawiła się jego ostatnia linia! Aby je zobaczyć trzeba było patrzeć nie — prosto w ekran, ale... od góry, prawie pionowo w dół! W trakcie dalszego przesuwania bliźniak poka-

zywał swe dalsze linijki, aż do momentu, kiedy do komórki 53248 ($d000) odpowiadającej za pionowe położenie duszka wstawiłem 55. Wtedy zniknął bez śladu. Dalsze przesuwanie nie przynosiło sensacyjnych efektów aż do momentu, kiedy 255 zamieniliśmy na 0. Wtedy na górze znowu pojawił się bliźniak, tym razem złożony od razu z ośmiu linii! Coś mi się zdaje, że umieszczanie duszków na obwódce nie będzie takie proste, jak by się zdawało. W nienormalnych (bo w końcu — czy jest normalny brak obwódki?) warunkach komputer „odwdzięcza się" nam nienormalną pracą.

Nie ma jednak problemów nie do rozwiązania. Wpiszcie, proszę, PROGRAM 3. Przepraszam, że jest taki długi, ale — tak po prostu wyszło. Zasada jego działania jest całkiem prosta, choć nieco skomplikowana. Jego zadaniem jest przesunięcie duszka w dół o jeden punkt oraz zlikwidowanie ramki i ewentualnych „bliźniaków". Kiedy duszek znajduje się między liniami 56 ($38) i 255 ($ff) niepotrzebne są żadne dodatkowe udziwnienia. Po przejściu zera musimy uruchomić drugie przerwanie na samym dole ekranu, zerujące komórkę 53269 ($d015) i — dzięki temu — likwidujące wszystkie duszki, które chciałyby się pokazać na górze. Gdy dojdziemy do linii 35, kiedy duszek jest już całkowicie niewidoczny, do komórki jego położenia pionowego 53249 ($d001) musimy znowu wstawić 0, tym razem zezwalając na wyświetlenie jedynie górnego bliźniaka. Proste? W każdym razie — to jedyny znany mi sposób na uniknięcie wyświetlania „bliźniaków". Jeśli znasz lepszy, niech przyśle go do redakcji — na pewno go opublikujemy!

Na zakończenie dzisiejszego odcinka chciałbym Wam zaproponować krótki, acz skuteczny program ściśle związany z wałkowanym dziś tematem. Ale sza! Wpisz najpierw i uruchom PROGRAM 4. I co? Nic? Tak! O to właśnie chodziło! Na oko nic się nie dzieje, lecz... Wpisz kilka POKE ukazujących nam duszka:

POKE 53248,160 : POKE 53269,1 : POKE 2040,0

No i co? Na obwódce mamy duszka! Nie likwidujemy obwódki całkowicie, a jedynie zmuszamy ją do chowania się pod duszkami? — Nie, to niemożliwe. No to może klasycznie ją kasujemy, lecz po prostu w odpowiedniej chwili zmieniamy kolor tła na taki sam, jak ma obwódka? Tak! Trzeba przyznać, że jest to skuteczne, a na nieprzygotowanych robi piorunujące wrażenie o czym donosi —

BARTEK I. KACHNIARZ

```
;PROGRAM 1

.:2710 78 a9 01 8d 1a d0 a9 7f
.:2718 8d 0d dc a9 31 8d 14 03
.:2720 a9 27 8d 15 03 a9 f7 8d
.:2728 12 d0 a9 1b 8d 11 d0 58
.:2730 60 ee 19 d0 a9 f9 cd 12
.:2738 d0 d0 fb a9 14 8d 11 d0
.:2740 a2 1b a9 ff cd 12 d0 d0
.:2748 fb 8e 11 d0 a9 5e 8d 14
.:2750 03 a9 27 8d 15 03 a9 30
.:2758 8d 12 d0 4c 81 ea ee 19
.:2760 d0 a9 f7 8d 12 d0 a9 1b
.:2768 8d 11 d0 a9 31 8d 14 03
.:2770 a9 27 8d 15 03 4c 31 ea
.:2778 d0 4c 81 ea ee 19 d0 ad
```

```
198 REM PROGRAM 2
199 :
200 POKE 53269,1
205 POKE 53248,160
210 POKE 2040,0
215 FOR A=0 TO 255
220 POKE 53249,A
225 PRINT A
230 GET A$: IF A$="" GOTO 230
235 NEXT
240 GOTO 215
```

```
; PROGRAM 3

.:2710 78 a9 01 8d 1a d0 a9 7f
.:2718 8d 0d dc a2 41 a0 27 a9
.:2720 f9 20 e6 27 a9 14 8d 11
.:2728 d0 a9 a0 8d 00 d0 a9 37
.:2730 8d 01 d0 a9 01 8d 15 d0
.:2738 a9 00 8d f8 07 58 4c 3e
.:2740 27 ee 19 d0 20 04 28 ee
.:2748 01 d0 ad 01 d0 c9 ff d0
.:2750 0d a2 7e a0 27 a9 f0 18
.:2758 20 e6 27 4c 81 ea ae 01
.:2760 d0 e0 3a b0 0c 8a a2 99
.:2768 a0 27 18 20 e6 27 4c 81
.:2770 ea a9 f0 8d 12 d0 a9 01
.:2778 8d 15 d0 4c 81 ea ee 19
.:2780 d0 ee 01 d0 a9 01 8d 15
.:2788 d0 20 04 28 a2 be a0 27
.:2790 a9 37 38 20 e6 27 4c 81
.:2798 ea ee 19 d0 a9 01 8d 15
.:27a0 d0 ad 12 d0 18 69 15 cd
.:27a8 12 d0 d0 fb a9 00 8d 15
.:27b0 d0 a9 f0 18 a2 41 a0 27
.:27b8 20 e6 27 4c 81 ea ee 19
.:27c0 d0 a9 00 8d 15 d0 ad 01
.:27c8 d0 c9 37 f0 0d a2 7e a0
.:27d0 27 a9 f0 18 20 e6 27 4c
.:27d8 81 ea a9 00 8d 01 d0 a2
.:27e0 41 a0 27 4c d1 27 8d 12
.:27e8 d0 90 0a a9 80 0d 11 d0
.:27f0 8d 11 d0 d0 08 a9 7f 2d
.:27f8 11 d0 8d 11 d0 8e 14 03
.:2800 8c 15 03 60 ad 11 d0 49
.:2808 08 8d 11 d0 a9 f9 cd 12
.:2810 d0 d0 fb ad 11 d0 49 08
.:2818 8d 11 d0 60 00 00 00 00
```

```
;PROGRAM 4

.:2710 78 a9 01 8d 1a d0 a9 7f
.:2718 8d 0d dc a9 31 8d 14 03
.:2720 a9 27 8d 15 03 a9 f7 8d
.:2728 12 d0 a9 1b 8d 11 d0 58
.:2730 60 ee 19 d0 ad 21 d0 8d
.:2738 b2 27 a9 00 8d ff 3f a9
.:2740 f9 ac 20 d0 cd 12 d0 d0
.:2748 fb a9 14 8d 11 d0 a9 fb
.:2750 cd 12 d0 d0 fb 8c 21 d0
.:2758 a2 9b a9 ff cd 12 d0 d0
.:2760 fb 8e 11 d0 a9 35 8d 12
.:2768 d0 a9 7c 8d 14 03 a9 27
.:2770 8d 15 03 ad 20 d0 8d 21
.:2778 d0 4c 81 ea ee 19 d0 ad
.:2780 b2 27 a8 a9 32 cd 12 d0
.:2788 d0 fb ad 12 d0 cd 12 d0
.:2790 f0 fb 8c 21 d0 a9 31 8d
.:2798 14 03 a9 27 8d 15 03 a9
.:27a0 f7 8 d12 d0 a9 1b 8d 11
.:27a8 d0 ad b2 27 8d ff 3f 4c
.:27b0 31 ea 00 18 a2 41 a0 27
```

6526 COMPLEX INTERFACE ADAPTER#1

— dokończenie

ze strony 30

PROGRAM 1

```
.,C000 A9 FF     LDA #$FF
.,C002 8D 02 DC  STA $DC02
.,C005 A9 00     LDA #$00
.,C007 8D 03 DC  STA $DC03
.,C00A A9 FE     LDA #$FE
.,C00C 8D 00 DC  STA $DC00
.,C00F AD 01 DC  LDA $DC01
.,C012 C9 FE     CMP #$FE
.,C014 D0 EA     BNE $C000
.,C016 60        RTS
```

PROGRAM 2

```
10 POKE 56322,255
20 POKE 56323,0
30 POKE 56320,254
40 IF PEEK(56321)=254 THEN STOP
50 GOTO 30
```

PROGRAM 3

```
10 POKE 56320,0
20 GOTO 10
```

PROGRAM 4A

```
.,C000 78        SEI
.,C001 A9 0E     LDA #$0E         ;BITY 0-4 CIAPRA
.,C003 2D 02 DC  AND $DC02        ; JAKO WEJSCIE
.,C006 8D 02 DC  STA $DC02
.,C009 A9 10     LDA #$10         ;TEST PRZYCISKU
.,C00B 2C 00 DC  BIT $DC00        ; FIRE
.,C00E D0 FB     BNE $C00B
.,C010 AD 00 DC  LDA $DC00
.,C013 29 0F     AND #$0F
.,C015 85 02     STA $02
.,C017 A2 08     LDX #$08         ;WYSWIETL 8 BITOW
.,C019 06 02     ASL $02
.,C01B A9 00     LDA #$00
.,C01D 69 30     ADC #$30
.,C01F 20 D2 FF  JSR $FFD2        ;PROCEDURA CHROUT
.,C022 CA        DEX
.,C023 D0 F4     BNE $C019
.,C025 A9 0D     LDA #$0D         ;CRT
.,C027 20 D2 FF  JSR $FFD2
.,C02A 4C 01 C0  JMP $C001
.,C02D
```

PROGRAM 4B

```
200 FOR A=49152 TO 49196
210 :READ B
220 :POKE A,B
230 :SK=SK+B
240 NEXT A
250 READ SU
260 IF SU<>SK THEN PRINT "ERROR!"
    :STOP
270 PRINT "OK!"
280 SYS 49152
290 DATA 120,169,224,045,002,220,
    141
300 DATA 002,220,169,016,044,000,
    220
310 DATA 208,251,173,000,220,041,
    015
320 DATA 133,002,162,008,006,002,
    169
330 DATA 000,105,048,032,210,255,
    202
340 DATA 208,244,169,013,032,210,
    255
350 DATA 076,001,192,5234
```



DRODZY CZYTELNICY!

Cieszę się bardzo, że do redakcji nadchodzi dużo zamówień zarówno na dyskietki jak i na archiwalne numery C&A. Oznacza to, iż nasze pismo cieszy się nie słabnącym zainteresowaniem. Ze swej strony staram się realizować zamówienia natychmiast po ich otrzymaniu. Niemniej czasem mogą zdarzyć się pewne opóźnienia, wynikłe z przyczyn losowych, za które serdeczenie przepraszam prosząc Was jednocześnie o wyrozumiałość.

Mimo najszczerszych chęci nie mogę jednak zrealizować tych zamówień, na których z powodu przeoczenia brak jest niektórych danych. Najczęściej zapominacie podać, które dyskietki/numery archiwalne chcielibyście kupić, bądź też podajecie niepełny adres. Oprócz tego niektórzy Czytelnicy zamawiają dyskietki 3.5" na Amigę, choć do tej pory nie drukowaliśmy przecież żadnych programów na ten komputer (dyskietki na Amigę będziemy sprzedawać począwszy od numeru 06/92). Często przysyłacie też najpierw zamówienie, a potem (w innym liście) potwierdzenie wpłaty. Nie muszę chyba mówić, że takie postępowanie niesłychanie opóźnia realizację zamówień...

Choć wiele spośród niechlujnie lub błędnie wypełnionych kuponów da się jakimś cudem odczytać to jednak Sherlockiem Holmesem nie jestem, i dlatego proszę następujących Czytelników o pilne skontaktowanie się z redakcją:

Marcin Daszkiewicz, Zgorzelec, ul. Wyspiańskiego ???
Dariusz Matysiuk, Siemianówka, ul. ???, kod pocztowy ???
Marcin Miłosz, Lublin, ul. ???
Waldemar Olszewski, Wałcz, ul. Żeromskiego 30/2
Krzysztof Siedlecki, Kołobrzeg, ul. Wąska 2/19
Jadwiga Suliborska, Łódź, ul. L. Okulickiego 16a/1
Marcin Górniak, Łańcut, ul. Solskiego 11a
Piotr Bogucki, Malcanów 27, kod pocztowy ???

A wszystkich Czytelników, którzy właśnie zamierzają coś u nas zamówić, proszę o WYRAŹNE wypełnianie kuponów z podaniem PEŁNEGO adresu i kodu pocztowego (zwłaszcza Czytelnicy mieszkający w małych miejscowościach). Jeśli dokonujecie wpłaty na innym blankiecie niż zamieszczony w C&A, wpisujcie DOKŁADNIE: **numer naszego konta** (zwykle „połykacie" jedno zero), **nazwę banku wraz z adresem** oraz **właściciela, którym jest Spółdzielnia „Bajtek" a nie „C&A"!** Jednocześnie informuję, że NIEPRAWIDŁOWO WYPEŁNIONE PRZEKAZY NIE BĘDĄ REALIZOWANE, A „PRZYJEMNOŚCI" ZWIĄZANE Z ODZYSKANIEM PIENIĘDZY SPADNĄ NIESTETY NA NIEDBAŁEGO NADAWCĘ!. Dla przypomnienia podaję więc PRAWIDŁOWE DANE jakie powinny znaleźć się na każdym przekazie:

Właściciel konta:
Spółdzielnia „Bajtek", ul. Wspólna 61, 00-687 Warszawa

Nazwa i adres naszego banku:
Bank „Agrobank S.A.", ul. Grochowska 262, 04-398 Warszawa
Nr konta: **470005-1834-131**

Kupon ważny do 09.08.1992

ZAMAWIAM PRENUMERATĘ:

( ) MIESIĘCZNIKA "C&A"
( ) MIESIĘCZNIKA "BAJTEK"
( ) DWUMIESIĘCZNIKA "TOP SECRET"

ZAMAWIAM NUMERY "C&A":

* 01/92: () magazyn () dyskietka
* 02/92: () magazyn () dyskietka
* 03/92: () magazyn () dyskietka
* 04/92: () magazyn () dyskietka
* 05/92: () magazyn () dyskietka
* 06/92: () magazyn () dyskietka



Kupon ważny do 09.08.1992

ZAMAWIAM PRENUMERATĘ:

( ) MIESIĘCZNIKA "C&A"
( ) MIESIĘCZNIKA "BAJTEK"
( ) DWUMIESIĘCZNIKA "TOP SECRET"

ZAMAWIAM NUMERY "C&A":

* 01/92: () magazyn () dyskietka
* 02/92: () magazyn () dyskietka
* 03/92: () magazyn () dyskietka
* 04/92: () magazyn () dyskietka
* 05/92: () magazyn () dyskietka
* 06/92: () magazyn () dyskietka



Na wszystkich odcinkach blankietu należy wpisać CZYTELNIE atramentem, długopisem lub pismem maszynowym kwotę cyframi oraz słownie, imię i nazwisko oraz adres wpłacającego.



Języki programowania komputerów czyli jak dogadać się z maszyną

Języki programowania są tak stare jak komputery, a nawet starsze. Pierwszy czysto teoretyczny język programowania PLANDAKUL został stworzony przez niemieckiego matematyka i inżyniera Kondrada Zuse w 1945 roku, natomiast pierwszy komputer — ENIAC — został zbudowany w 1946 roku. Język ten, mimo iż nigdy nie doczekał się swojej realizacji na komputerze, uważany jest za praojca języków programowania wysokiego poziomu.

Powstawanie współczesnych języków programowania datuje się od połowy lat pięćdziesiątych, kiedy to w roku 1954 opracowano pierwszy język wysokiego poziomu — FORTRAN. Do programowania komputerów stosowano język wewnętrzny, zwany inaczej maszynowym. Jak ogólnie wiadomo, jedynym zrozumiałym językiem dla komputera jest ciąg liczb. Ponieważ wyszukiwanie i poprawianie ewentualnych błędów w programie liczącym kilkaset liczb było żmudne i pracochłonne, wymyślono języki symboliczne, w których każdej instrukcji odpowiada logicznie związany z nią symbol. Do tego dodano etykiety (symbole adresów), polecenia przypisywania wartości i rezerwacji pamięci, oraz komentarze, i tak powstały asemblery. Początkowo nazwa „asembler" odnosiła się do języka programowania, natomiast program tłumaczący kod źródłowy na język wewnętrzny nosił nazwę translatora. Obecnie mianem asemblera określa się nie tylko język ale i translator, edytor, monitor pamięci i ostatnie „dziecko" programistów — source debugger, czyli program do śledzenia działania programu w postaci źródłowej.

Języki te z powodu ścisłego związku z budową danego komputera noszą nazwę języków programowania niskiego poziomu. Mają one swoje wady i zalety. Niewątpliwymi zaletami jest mała objętość programu wynikowego i jego szybkość. Zaś główną wadą jest to, że nie można przenosić programu napisanego dla jednego komputera na inny. Poza tym programowania w tych językach wymaga, oprócz znajomości listy rozkazów, również tzw. mapy pamięci oraz znajomości architektury konkretnego komputera. I ostatnia wada: mimo zastosowania symboli programy napisane w asemblerze są mało czytelne.

Wad tych pozbawione są języki wysokiego poziomu, czyli takie, które pozwalają na przedstawienie zadania w formie wzoru matematycznego lub problemu logicznego. Programy tak napisane mogą być prawie całkowicie przenoszone z komputera na komputer. To „prawie" wiąże się z ewolucją danego języka i zabiegami programistów mającymi na celu jak najlepsze dopasowanie języka do komputera, co ułatwia pełne wykorzystanie możliwości maszyny. Języki te są używane do prezentacji algorytmów w publikacjach oraz do nauki informatyki. Przejrzystość i prostota układania programów zostały okupione objętością i szybkością działania programów wynikowych.

FORTRAN (FORmula TRANslation)

Pierwszym z tego rodzaju języków jest FORTRAN. Został stworzony głównie do obliczeń numerycznych, czego dowodem jest brak do wersji 77 zmiennej typu znakowego (miał za to zmienną typu zespolonego czyli „complex"). Napisanie w roku 1957 kompilatora FORTRANU otworzyło nowy rozdział w programowaniu. Swą dużą popularność, obecnie nieco gasnącą, FORTRAN zawdzięcza olbrzymiej liczbie funkcji bibliotecznych, dodawanych na przestrzeni lat i obejmujących między innymi wszystkie ważniejsze algorytmy obliczania całek, odwracania macierzy czy rozwiązywania układów równań.

ALGOL (ALGOrithmic Language)

Następnym, chronologicznie, językiem jest ALGOL. Pierwsze prace nad tym językiem podjęli Niklaus Wirth i K.A.R. Hoare w 1958 roku. Pierwszy kompilator Algolu powstał w 1960 roku. Podobnie jak FORTRAN, ALGOL jest językiem przeznaczonym głównie do obliczeń, ma jednak większe od FORTRANU możliwości, dzięki czemu stał się pierwszym językiem dydaktycznym. ALGOL przegrał niestety w konkurencji z FORTRANEM: ostatnia wersja powstała w roku 1968 (zaś FORTRANU w 1977).

COBOL (COmmon Business Oriented Language)

Do nieco innych celów napisany został kolejny język, COBOL. Stosowany jest on do pisania programów dla gospodarki, handlu i bankowości. COBOL pozwala na przeprowadzanie tylko prostych obliczeń, za to potrafi przetwarzać olbrzymie ilości danych. W odróżnieniu od poprzedników składania tego języka jest zbliżona do języka naturalnego (mówionego) co znacznie ułatwia czytanie programów, ale niekoniecznie ułatwia ich analizę.

BASIC (Beginner's All Purpose Symbolic Instruction Code)

Pierwszą wersję tego języka opracowali w 1964 roku J.G. Kemmeny i T.F. Kurtz. Zamiarem jego twórców było stworzenie uniwersalnego języka dla początkujących programistów. Język ten jako pierwszy pozwalał na pracę w trybie bezpośrednim, bowiem przeważnie był interpreterem. Instrukcje programu przed wykonaniem są sprawdzane pod względem składni, co ułatwia początkującym poprawienie błędów powstałych przy wprowadzaniu programu. Tryb bezpośredni pozwala też na natychmiastowe sprawdzenie efektu działania pojedyńczej instrukcji (oczywiście nie dotyczy to pętli). Gwałtowny rozwój BASIC nastąpił z chwilą zastosowania go w pamięciach ROM mikrokomputerów. Rozwój ten był tak gwałtowny, że nie doczekał się standardu. Jest jedynym językiem wysokiego poziomu, który stosunkowo trudno przenosi się z komputera na komputer, gdyż jego dialekty bardzo się różnią między sobą. Najczęściej wymienianą wadą tego języka jest niestrukturalność, prowadząca do złych nawyków w sztuce programowania; w BASIC brak także procedur i zmiennych lokalnych (choć wyłom w tym zakresie zrobił m.in. TURBO BASIC firmy Borland przeznaczony dla komputerów PC), a obecnie dostępne interpretery czy kompilatory są już tych wad pozbawione.

Pascal

Kolejnym, bardzo popularnym językiem, który chciałbym przedstawić jest Pascal. Jego autorem jest Niklaus Wirth, twórca ALGOLU. Niewątpliwie swój sukces Pascal zawdzięcza opracowaniu wersji dla komputerów PC (Turbo Pascal). Obecnie jest on najczęściej stosowanym językiem dydaktycznym. Pozwala on na pełne programowanie strukturalne, to znaczy takie w którym program buduje się z „modułów". W efekcie program jest przejrzysty, co wynika również z wymuszonej budowy programu, w której część operacyjna jest oddzielona od części deklaracyjnej. Ponieważ każda zmienna w Pascalu musi być zadeklarowana przed użyciem, programista musi dobrze przemyśleć sens jej użycia. Cechą charakterystyczną Pascala jest nowy typ zmiennych: Record i Plik, świetnie nadających się do pisania baz danych. Kolejnym plusem są bardzo szybkie kompilatory, oraz „source debuggery". Oczywiście te możliwości mają tylko niektóre wersje tego języka.

MODULA 2

Ostatnim z tej grupy jest język o nazwie MODULA 2. Jest on „pascalopodobny", (ułożył go zresztą twórca Pascala). MODULA jest językiem pozwalającym na łączenie niezależnych modułów, napisanych przez różnych ludzi (lub zespoły) w jedną funkcjonalną całość, co przy opracowywaniu dużych programów jest rzeczą konieczną. Poza tym ułatwia on programowanie systemów wieloprocesorowych i wielozadaniowych (piękne pole do popisu ma tutaj Amiga). W przyszłości MODULA być może zastąpi Pascala

(cdn)

PAWEŁ GALAS (THEAM ROBOCOST)

TOP SECRET

MAGAZYN FANÓW GIER KOMPUTEROWYCH
MAJ CZERWIEC
'92

cena 10000

10/3

MEGA LO MANIA

MIECZE VALDGIRA
THE LORD OF THE RINGS
WING COMMANDER II

F-29 RETALIATOR
KONKURS SF

- już w kioskach!

Index 379859